IMPROVED EXACT METHODS FOR STATISTICAL INFERENCE IN CONTINGENCY TABLES

By

DONGUK KIM

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

1994

To my wife, daughter

and

my parents

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Dr. Alan Agresti. Without his guidance and encouragement, this work would not have been completed. I would like to thank Dr. Mark Yang, Dr. Myron Chang, Dr. Brett Presnell, and Dr. David Wilson for their encouragement and advice while serving on my dissertation committee.

In my six years as a student here, I learned from all professors. I would also like to thank Dr. Yang and Dr. Randles for all the support while I worked as a consultant in the Biostatistics Division and as a teaching assistant. Also my thanks go to all my colleagues and friends.

Finally, I wish to express my special thanks to my family, especially my wife, YoungHee, for her love, patience, and encouragement, and my daughter, Minjee for her love. Furthermore, I would like to thank my parents for their love, encouragement, and support.

TABLE OF CONTENTS

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment
of the Requirements for the Degree of
Doctor of Philosophy

IMPROVED EXACT METHODS FOR STATISTICAL INFERENCE
IN CONTINGENCY TABLES

By

Donguk Kim

August 1994

Chairman: Alan Agresti
Major Department: Statistics

Ordinary "exact" methods can be highly conservative when the distribution of the test statistic is discrete. This becomes more severe as the number of dimensions or the number of categories is small. We improve exact inferential methods by decreasing the conservativeness that occurs due to discreteness. In this dissertation, modifications of exact inferential methods are suggested for conditional associations in three-way contingency tables. For testing conditional independence, we present a modified P-value. It utilizes both the usual test statistic and, at the observed value of that statistic, a supplementary statistic directed toward a broader alternative. For $2 \times 2 \times K$ tables, we propose modified "exact" confidence intervals for an assumed common odds ratio based on inverting two separate one-sided tests using the modified P-value. We also present an alternative and usually even better way of constructing "exact" confidence intervals, based on inverting a two-sided test with a modified P-value.

For $I \times J \times K$ tables, we discuss exact tests of conditional independence using six test statistics that have connections with loglinear models. Three statistics assume a lack of three-factor interaction, and the other three statistics do not require this

assumption. All six statistics are score statistics for loglinear models that treat none, one, or both of the classifications as ordinal. Then, we discuss possible alternative ways of forming modified exact P-values in $I \times J \times K$ contingency tables, and we propose modified exact P-values for six tests corresponding to six loglinear models. For three-way contingency tables, computational algorithms have limited availability for tests of conditional independence when I and J exceed two. We use a simulation algorithm to obtain precise estimates of ordinary and modified exact P-values for cases for which the current computational algorithms are infeasible.

For $I \times J \times K$ tables, we show how to construct exact, unbiased, and admissible tests for an ordinal alternative to conditional independence by using a modified P-value approach. This is a generalization of the results of Cohen and Sackrowitz for a test of independence in two-way contingency tables for an ordinal alternative. The ordinary test of conditional independence for $2 \times 2 \times K$ contingency tables is usually inadmissible.

CHAPTER 1
INTRODUCTION

1.1 Literature Review

Statistical inference for contingency tables generally is carried out by large-sample approximations for sampling distributions of the test statistic rather than the exact discrete distribution. A central concern is the quality of the asymptotic approximation. Large-sample approximations apply as the sample size grows, for a fixed number of cells. The adequacy of the chi-square approximation depends on both the sample size and the number of cells. Some contingency tables occur where the sample size is too small to apply asymptotic methods. Also, high-dimensional contingency tables tend to be sparse, and as a consequence the asymptotic approximation to the sampling distribution is often very poor. Agresti (1992) surveyed exact inference for contingency tables and explained the developments of exact methods for contingency tables. He suggested the use of exact methods instead of large-sample approximations when the application of asymptotic approximation is questionable. We focus on exact inferential methods for conditional associations in three-way contingency tables.

When the exact distribution of the test statistic is discrete, it is known that ordinary "exact" tests and confidence intervals can be highly conservative because of the discreteness of the distribution. Though exact tests are guaranteed to control the probability of Type I error at any nominal level, we may not achieve a probability of Type I error of the nominal level exactly. The actual probability of Type I error may be considerably smaller. For instance, in a 2×2 contingency table, Fisher's

exact test is always conservative. For exact inference about a parameter of interest, we condition on sufficient statistics for unknown parameters to eliminate them. For an exact conditional test for categorical data, the reference set of tables over which the exact conditional distribution is defined is the set of contingency tables having certain marginal counts fixed. This extra conditioning makes the distribution of the test statistic more highly discrete.

Barnard (1947) proposed an unconditional exact test for 2×2 contingency tables. The reference set of his test is defined as the set of all tables with fixed row margins and all possible column margins. Since the column margins are not fixed, this unconditional test has many more tables in the reference set, and the distribution of the test statistic is less discrete. A disadvantage of the unconditional test is that computations are infeasible for larger tables, since maximizing over the space of nuisance parameters is needed for implementation. For further details, see Yates (1984) and Suissa and Shuster (1985).

One way to reduce conservativeness is the mid P adjustment. Let T be a test statistic and t_o be its observed value. According to Lancaster (1961), the mid P adjustment utilizes half of the probability of the observed value of T; hence, it subtracts half of the probability of the observed statistic from the usual exact P-value. This reduces the conservativeness due to discreteness and does not rely on randomization to eliminate the conservativeness. But one drawback is that it can not guarantee exactness, in the sense that the actual size possibly exceeds the nominal level. It comes from the fact that the mid P approach subtracts half of the probability of the observed statistic from the exact P-value.

For nonparametric tests, Streitberg and Roehmel (1990) considered utilizing a secondary statistic together with the usual statistic to discriminate among those rank configurations that have the same value of the primary statistic. He showed that his test is uniformly more powerful than the Wilcoxon-Mann-Whitney test, and the

P-value of this test employing any secondary statistic can not be larger than the P-value from the ordinary test. A similar approach to reduce the conservativeness is due to Cohen and Sackrowitz (1992). They suggested a modified P-value that utilizes both the usual test statistic and, at the observed value of that statistic, the null table probability for a secondary partitioning for those tables having $T = t_o$. Instead of including all tables having $T = t_o$ in the calculation of the P-value, they include tables that are no more likely than the observed. They used this for ordinal tests in two-way tables.

Discreteness also affects interval estimation. An "exact" confidence interval for a parameter can be constructed by inverting the exact conditional test. The ordinary confidence interval (Cox 1970, Gart 1970, Mehta *et al.* 1985, Vollset *et al.* 1991) is based on inverting two separate one-sided tests using the ordinary P-value. Because of discreteness, we get a conservative confidence interval. The actual confidence coefficient is at least the nominal level.

We could construct an exact confidence interval based on inverting a single two-sided test rather than two separate one-sided tests. Using a two-sided approach, Sterne (1954) constructed a confidence interval for a single binomial parameter, and Baptista and Pike (1977) constructed confidence limits for the odds ratio in a 2×2 table. This two-sided confidence interval also is conservative.

Some problems arise when exact methods are infeasible and the application of large-sample approximations is questionable. For large-sample inference about conditional association in three-way contingency tables, Mantel and Haenszel (1959) gave a test statistic comparing two groups on a binary response, adjusting for control variables. Since Cochran (1954) proposed a similar statistic, it is called the Cochran-Mantel-Haenszel statistic. This is a test for conditional independence in $2 \times 2 \times K$ tables. Also, Birch (1964) showed that under the assumption of a constant odds ratio within each of the tables, this test is uniformly most powerful unbiased.

Birch (1965) derived three test statistics for testing the null hypothesis of conditional independence of two variables in $I \times J \times K$ contingency tables. These are score statistics for loglinear models that none, one, or both of the classifications are ordinal. These models assume a lack of three-factor interaction. When both classifications are nominal, the corresponding statistic is a generalized Cochran-Mantel-Haenszel test statistic to handle more than two groups or more than two responses. This method involves computing the expected values and the covariance matrix under the multiple hypergeometric probability model for each of the tables. These quantities then are summed across the tables, and a quadratic form of the test statistic is generated. When both classifications are ordinal, the corresponding statistic is the same as Mantel's (1963) score statistic. Furthermore, Birch's statistics are special cases of a general statistic proposed by Landis *et al.* (1978). These statistics have an asymptotic chi-squared distribution.

Rather than use large-sample approximations, we wish to conduct exact inference. Even though recent developments make exact methods feasible for some inferential analyses, because of computational complexity, we do not have exact methods for some situations. For three-way contingency tables, current computational algorithms for exact methods are restricted to certain analyses for $2 \times J \times K$ tables with ordered columns.

The Monte Carlo method is another alternative to either exact or asymptotic methods. This method is based on estimating the exact conditional sampling distribution of the statistic by generating random tables having the relevant fixed margins. It is useful for those situations where the data set is too large for an exact computation or too sparse to rely on the asymptotic theory. For table generation by simulating from a hypergeometric distribution, Boyett (1979) wrote a program that generates a two-way random table from the exact distribution with given row and column totals.

Patefield (1981) presented a program generating a random table, and his program is faster than Boyett's for larger sample sizes.

Agresti *et al.* (1979) utilized the Monte Carlo method effectively for a variety of tests for two-way tables. Even for large tables or large sample sizes, one can quickly approximate as closely as needed the ordinary and modified exact P-values for these statistics. This method consists of sampling contingency tables from the conditional reference set in proportion to their probabilities and computing an unbiased point estimate and a narrow confidence interval for an exact P-value.

When we construct a critical region for exact tests with some preassigned nominal level α, supplementary randomization would be required at the boundary of the critical region in order to achieve the nominal size. This is typical for any discrete problem. After randomization, the resulting test may be inadmissible. Cohen and Sackrowitz (1991) focused on two-way tables and showed unbiasedness for the test of independence in two-way tables for an ordinal alternative. Eaton (1970) showed the essentially complete class in an exponential family. Eaton's theorem shows that the essentially complete class consists of tests whose acceptance regions are convex with possible randomization on the boundary of acceptance region. Furthermore, Ledwina (1978a, 1984) gave the class of admissible rules in an exponential family. Using the same argument in Ledwina, Cohen and Sackrowitz (1991) proved a theorem that gives the class of exact, unbiased, and admissible tests in two-way contingency tables. They constructed the exact test of size α by ordering the tables according to their probabilities on sample points where the test would randomize. They made the number of tables on which randomization would occur considerably smaller than in the usual test.

1.2 Summary of Dissertation Work

In Chapter 2, we present exact tests of conditional independence against the alternative of no three-factor interaction. Our modified exact tests are adaptations of the ordinary exact conditional tests that are less conservative. We propose a modified P-value based on a secondary partitioning of the sample space beyond that generated by the test statistic. It utilizes both the usual test statistic and, at the observed value of that statistic, a supplementary statistic T' directed toward a broader alternative. In the calculation of the P-value, we include only those tables that are at least as contradictory to the null in terms of T'. One can calculate this modified P-value for any test statistic having a discrete distribution. The modified P-value is less discrete than the ordinary P-value, does not employ randomization, and leads to a less conservative "exact" test.

By inverting results of tests using modified P-values, we obtain an exact and less conservative confidence interval, in the sense that the modified confidence interval has confidence coefficient at least the nominal level and is narrower than the ordinary one. For $2 \times 2 \times K$ tables, we suggest a modified "exact" confidence interval inverting the test based on a modified one-sided P-value to make the actual confidence coefficient closer to the nominal value. Also, we present an alternative and usually even better way of constructing "exact" confidence intervals, based on inverting a two-sided test with a modified P-value.

Furthermore, we utilize the mid P-value to construct intervals applying these methods, although these are not exact. To compare these types of intervals, we calculate actual coverage probability or expected length of the confidence intervals based on inverting one-sided or two-sided tests using the ordinary or modified P-value.

In Chapter 3, we suggest exact inference regarding conditional associations in three-way contingency tables. For exact tests of conditional independence in $I \times J \times K$ tables, three statistics assuming a lack of three-factor interaction are discussed, and then we provide three other test statistics permitting three-factor interaction. All six test statistics are score statistics for loglinear models that treat none, one, or both of the classifications as ordinal. Also they have asymptotic chi-squared distributions. Using these statistics, we propose modified exact P-values for six tests for testing conditional independence with $I \times J \times K$ tables.

For cases that are currently computationally infeasible, we construct a simulation algorithm to obtain precise estimates of ordinary and modified exact P-values, using a table-generation procedure suggested by Patefield (1981). We utilize six test statistics for exact tests of conditional independence.

In Chapter 4, we generalize results of Cohen and Sackrowitz (1991, 1992) to construct exact, unbiased, and admissible tests for an ordinal alternative to conditional independence for $I \times J \times K$ tables. We first show unbiasedness of tests when one wishes to test a null hypothesis of conditional independence against the alternative of no three-factor interaction model in three-way contingency tables. Then we present the complete class of tests and admissible tests in an exponential family following Eaton (1970) and Ledwina (1978a, 1984). Using these arguments, we generalize to the three-way case some results of Cohen and Sackrowitz regarding admissibility of tests for two-way tables. Combining these, we have a theorem that gives the class of exact, unbiased, and admissible tests in three-way contingency tables.

With this theorem, we discuss how to construct unbiased tests and how to set up critical regions to obtain tests of conditional independence of fixed size α, for an ordinal alternative. We construct the exact test of size α by ordering the tables according to a secondary statistic directed toward a broader alternative hypothesis at the randomization points, utilizing the modified approach discussed in Chapter 2. By

using the modified approach, the resulting test is admissible after randomization, and it requires less randomization than usual. Also, we have actual size closer to a nominal value. The Appendix contains a FORTRAN program. Using this program, one can easily get ordinary and modified exact inference about conditional associations for $2 \times 2 \times K$ contingency tables.

CHAPTER 2
IMPROVED EXACT INFERENCE ABOUT CONDITIONAL ASSOCIATION

2.1 Introduction

When a test statistic has a discrete distribution, ordinary "exact" tests and confidence intervals can be highly conservative due to discreteness. If we conduct a test using some preassigned size α, the probability of Type I error is always less than or equal to a preassigned value. If one constructs an "exact" confidence interval with confidence coefficient $1 - \alpha$, the actual confidence coefficient is *at least* that level and is unknown (Neyman 1935). We wish to improve ordinary exact inferential methods by decreasing the conservativeness that occurs due to discreteness. In this chapter, we suggest modifications of exact inferential methods for conditional associations in $2 \times 2 \times K$ contingency tables.

For instance, we present an example of a $2 \times 2 \times 5$ table for which the ordinary 95% confidence interval for an assumed common odds ratio is (1.1, 531.5). The discreteness implies that .95 is a lower bound for the actual confidence coefficient. We show how to construct a modified confidence interval that also has the guarantee of at least 95% confidence, but takes the much shorter range (2.1, 67.3). Our approach is applicable for any contingency table of size larger than 2×2, but we illustrate the arguments in terms of inferences about conditional associations in $2 \times 2 \times K$ contingency tables. The ideas and notations apply throughout the dissertation. In this chapter we are focusing on $2 \times 2 \times K$ contingency tables.

For three-way tables, consider the hypothesis of conditional independence of two variables, given the third one. For instance, if $\{\pi_{ijk}\}$ denote probabilities for a multinomial distribution over the $I \times J \times K$ cells, where $\Sigma\Sigma\Sigma\pi_{ijk} = 1$, the hypothesis states that

$$\pi_{ijk} = \pi_{i+k}\pi_{+jk}/\pi_{++k}.$$

The subscript "+" denotes the sum over the index it replaces. Let $\mathbf{N} = \{n_{ijk}\}$ denote the cell counts, with expected frequences $\{m_{ijk}\}$. We discuss exact conditional tests of this hypothesis, generalizing Fisher's exact test for 2×2 tables. We also discuss confidence intervals for odds ratios pertaining to conditional association.

Let X denote the row classification, Y the column classification, and Z the layer classification. The hypothesis of conditional independence of X and Y, given Z, is usually tested against the alternative of no three-factor interaction. This alternative is the loglinear model of form

$$\log m_{ijk} = \mu + \lambda_i^X + \lambda_j^Y + \lambda_k^Z + \lambda_{ij}^{XY} + \lambda_{ik}^{XZ} + \lambda_{jk}^{YZ}, \tag{2.1}$$

having sufficient statistics $(\{n_{ij+}\}, \{n_{i+k}\}, \{n_{+jk}\})$. The null hypothesis corresponds to the special case of this model in which all $\lambda_{ij}^{XY} = 0$. Exact conditional tests utilize the distribution of $\{n_{ij+}\}$, the sufficient statistics for these parameters, conditional on the other sufficient statistics, that relate to the remaining parameters. For the case of a $2 \times 2 \times K$ table, for instance, one uses the distribution of $\sum_k n_{11k}$, conditional on the row totals $\{n_{i+k}\}$ and column totals $\{n_{+jk}\}$ for the partial tables (Birch 1964). The parameter of interest for estimation is the assumed common odds ratio for each 2×2 table.

We present exact tests of conditional independence for the alternative of no three-factor interaction. Our modified exact tests are adaptations of the ordinary exact conditional tests that are less conservative. They use a modified P-value based on a secondary partitioning of the sample space beyond that generated by the test statistic. It utilizes both the usual test statistic and, at the observed value of that statistic,

a supplementary statistic directed toward a broader alternative. A modified P-value is less discrete than the ordinary P-value and leads to less conservative "exact" tests. By inverting results of tests using modified P-values, we have an exact and less conservative confidence interval, in the sense that a modified confidence interval has confidence coefficient at least the nominal level, and it is narrower than the ordinary one.

Section 2 introduces the modified P-value and shows that its distribution can be much less discrete than that of the ordinary P-value. We compare the ordinary and modified P-values with examples. Furthermore, the null expected value of the P-value is discussed in both procedures in order to examine the degree of conservativeness. Section 3 discusses modified "exact" confidence intervals, based on inverting two one-sided tests using the modified P-value. Though they are also conservative, they may be much narrower than the usual one. Illustrations are given for estimating an assumed common odds ratio for several 2×2 tables. Section 4 presents an alternative and usually even better way of constructing "exact" confidence intervals, based on inverting a two-sided test with a modified P-value. Section 5 discusses some related results for logistic regression models, and Section 6 gives some comments.

2.2 A Less Conservative P-value

Suppose we would like to conduct an exact conditional test for categorical data using some preassigned size α, such as 0.05. Denote by Γ the set of contingency tables having the same marginal counts as the ones that are fixed by the conditioning argument for the exact conditional test. This is the set of tables over which the exact conditional distribution is defined. For the test of conditional independence,

for instance, Γ is the set of $I \times J \times K$ tables of nonnegative integers, $\Gamma = \{\mathbf{Z} : \ \Sigma_i z_{ijk} = n_{+jk}, \ \ \Sigma_j z_{ijk} = n_{i+k}, \quad \text{for all } i, j, k\}$.

It is usually not possible to construct a critical region for exact conditional tests with preassigned size α because of the discreteness of the distribution. If an exact test is desired of arbitrary size α, supplementary randomization would be required to make the decision about whether to reject when a table occurs at the boundary of the critical region. In practice, it is unacceptable to employ randomization, and one normally simply reports a P-value. In general, suppose we have a test statistic T, such as a Wald, likelihood ratio, or score statistic, and suppose t_o is the observed value of T. If large values of T contradict the null, the usual P-value is

$$P = P_{H_0}(T \geq t_o), \tag{2.2}$$

the probability under the null hypothesis that T is at least t_o. Ordinarily, if one wants to make a decision about H_0, one rejects if the P-value $\leq \alpha$. The discreteness implies that the test based on the P-value is conservative in the sense that the actual size is

$$P_{H_0}(P \leq \alpha) \leq \alpha \ \text{ for } \ 0 < \alpha < 1. \tag{2.3}$$

In the exact conditional approach, one conditions on sufficient statistics for unknown parameters in order to eliminate them. Then, the tail probability that determines the P-value does not depend on unknown parameters and can be exactly calculated. The extra conditioning reduces the set of possible test statistic values, making the distribution more highly discrete. Hence, tests of nominal size α based on the exact conditional P-value can be even more conservative. The actual probability of Type I error can be considerably less than the nominal value unless the sample size is reasonably large. This problem is exacerbated by the tendency of many users to put too much emphasis on testing at sacred levels such as .05. One can argue that one should simply report the P-value and not make comparisons to such arbitrary levels,

particularly when data are discrete. However, the discreteness also affects interval estimation.

2.2.1 The Modified "Exact" P-value

To reduce the degree of conservativeness, we suggest a modified P-value based on a less discrete distribution than that of T. The modified P-value uses a partition of the sample space that is more refined than we get using T alone. We use T to construct a primary partitioning of all tables that have the sufficient statistics fixed by the conditional test. Then, within fixed values of T, we generate a secondary partitioning using some other index T' of the degree to which the data contradict the null hypothesis. The statistic T' is a test statistic directed toward a somewhat broader alternative hypothesis, hence detecting information that may be missed by T. Let t_o and t_o' denote the observed values of the primary and secondary statistic. The modified P-value is defined as

$$P^* = P_{H_0}(T > t_o) + P_{H_0}(T = t_o,\, T' \geq t_o'), \tag{2.4}$$

where the probabilities are computed under the null conditional distribution. Instead of including all tables having $T = t_o$ in the calculation of the P-value, we include only those that are at least as contradictory to the null in terms of having at least as large a value of T'.

To illustrate, consider testing conditional independence in $2 \times 2 \times K$ tables. Normally, if we expect about the same strength of association in each 2×2 stratum, we test against the alternative (2.1) of no three-factor interaction. Using this narrow alternative helps to build power compared to statistics based on the general alternative, even if we do not feel that reality exactly satisfies (2.1). Suppose we use as the primary statistic the score statistic, which is based on $T = \sum_k n_{11k}$, for the conditional

set of tables having the same row and column totals as the observed table. Then one could use the score statistic for the general alternative (the saturated model) for the secondary partitioning. This is simply $T' = \sum_k X_k^2$, where X_k^2 denotes the Pearson statistic for testing independence in the kth partial table. The secondary statistic also contains information about the validity of the null hypothesis, but is directed toward a wider alternative.

Another possibility for the secondary partitioning is to use the null table probability, in which case T' can be expressed as the negative log of that probability. For a given value of T, tables that are less likely under the null are then considered to give greater evidence against the null. Let $B = \{\mathbf{Z} : \mathbf{Z} \in \Gamma,\ T = t_o,\ P(\mathbf{Z}) \leq P(\mathbf{N})\}$, where the probabilities are computed under the null. The modified P-value is then

$$P_p^* = P_{H_0}(T > t_o) + P_{H_0}(B). \tag{2.5}$$

The modified P-value orders sample tables in Γ according to their probabilities when $T = t_o$. Hence, this is based on the probability of the observed table as well as some test statistic. Cohen and Sackrowitz (1992) used this type of P-value for ordinal tests in two-way tables. We will compare both ways of forming modified P-values and confidence intervals based on these modified P-values, with examples. We prefer P^* over P_p^* for the modified P-value, because both T and T' are score statistics for testing conditional independence.

The setting and the statistic T in definitions (2.4) and (2.5) are arbitrary. One can calculate P^* for any test statistic having a discrete distribution, since it satisfies $P_{H_0}(P^* \leq \alpha) \leq \alpha$ for $0 < \alpha < 1$. We show that under the null this modified P-value has the property,

$$P_{H_0}(P^* \leq \alpha) \leq \alpha \ \text{ for } \ 0 < \alpha < 1. \tag{2.6}$$

Let P^* be a modified P-value and let $\boldsymbol{m}$ be a possible marginal configuration. We first show that the conditional P-value has $P_{H_0}(P^* \leq \alpha|\boldsymbol{m}) \leq \alpha$. The result is

easily obtained by noting that the modified P-value is a special case of the usual P-value using a more refined partitioning of T and T'. The ordinary P-value uses a partitioning based on T, and it is the sum of $P_{H_0}(T = t_o)$ and the probability of more extreme values of T. The modified mid P-value uses a partitioning based on T and T' within T. Let Max($\cdot$) denote the maximum value, let Min($\cdot$) denote the minimum value, and let Gap(T) denote the minimum difference between two consecutive values of T. We assume that T and T' have positive values. Define a new statistic $T^* = T \times \text{Max}(T')/\text{Gap}(T) + T'$. If Min($T'$) equals 0, we transform from T' to $T' + 1$ in order to avoid ties in T^*. Then, $T^*(\mathbf{Z}_1) > T^*(\mathbf{Z}_2)$ for all tables $\mathbf{Z}_1, \mathbf{Z}_2$ with $T(\mathbf{Z}_1) > T(\mathbf{Z}_2)$. Let t_o^* denote the value of T^* for the observed table. Note that a partitioning of the sample space using T and T' within T is equivalent to a partitioning of the sample space using T^*. Since there are no ties, ordering tables using T and T' within T is equivalent to ordering tables using T^*. Then, the sum of the probability that T' is at least T_o' at $T = t_o$ and the probability of more extreme values of T is equivalent to the sum of $P_{H_0}(T^* = t_o^*)$ and the probability of more extreme values of T^*. That is,

$$\begin{aligned} P^* &= P_{H_0}(T > t_o) + P_{H_0}(T = t_o,\, T' \geq t_o') \\ &= P_{H_0}(T^* > t_o^*) + P_{H_0}(T^* = t_o^*). \end{aligned}$$

Hence, the modified P-value is a special case of the usual P-value with a more refined partitioning, and we have $P_{H_0}(P^* \leq \alpha|\boldsymbol{m}) \leq \alpha$. Then, under the null,

$$P_{H_0}(P^* \leq \alpha) = E[P_{H_0}(P^* \leq \alpha|\boldsymbol{m})] \leq \alpha, \tag{2.7}$$

since the average of these conditional modified P-values over all possible marginal configurations is less than or equal to α. Thus, we have shown that the probability of Type I error is no greater than the nominal value.

The modified P-values can not be larger than the ordinary P-values, so the test based on it is less conservative in the sense that the actual size is closer to the nominal

value. Also, the sampling distribution of the modified P-value is less discrete than usual in the sense that its support can have considerably more points. When each table with a particular statistic value T has the same value of T', then P^* is the same as the usual exact P-value. As a special case, when there is only one table having each distinct value of T, such as in Fisher's exact test, they are identical. Note that if T is a score or Wald or likelihood-ratio statistic for a particular alternative, it does not help to take T' to be one of the other statistics for that same alternative. Because these tests all depend only on the sufficient statistics under the alternative, two tables that have the same value of T also have the same value of T', when T and T' are taken from these procedures. Thus, we base T' on a more general alternative, for which the extra sufficient statistic provides a finer partitioning.

When a test statistic has a continuous distribution, the P-value has a uniform(0,1) null distribution. Hence, for the continuous case the expected value of P-value is $\frac{1}{2}$. We prove now that in the discrete case the expected value of P under the null is greater than $\frac{1}{2}$. For an arbitrary random variable X (Mood, Graybill and Boes 1974, page 65),

$$\begin{aligned} EX &= \int_0^\infty [1 - F_X(x)]dx - \int_{-\infty}^0 F_X(x)dx \\ &= \int_0^\infty [1 - \Pr[X \le x]]dx - \int_{-\infty}^0 \Pr[X \le x]dx. \end{aligned}$$

Thus, $EP = \int_0^1 [1 - \Pr[P \le p]]dp$. Since, from (2.6) $1 - \Pr[P \le p] \ge 1 - p, \ \ 0 < p < 1$, we have

$$EP \ \ge \ \int_0^1 [1 - p]dp = \frac{1}{2}.$$

In the discrete case, the P-value is stochastically larger than the uniform, and its expected value exceeds $\frac{1}{2}$. Hence, we can describe the degree of conservativeness by

comparing $E_{H_0}P$ to 0.5. If the expected value exceeds 0.5 by much, the conservativeness is severe.

2.2.2 The Modified Mid P-value

The mid P-value (Lancaster 1961) is another alternative to the usual P-value that many statisticians have recommended as a way of compromising between having a conservative test and using supplementary randomization (*e.g.*, Barnard 1990). It is defined by

$$P_{\text{mid}} = P_{H_0}(T > t_o) + (1/2)P_{H_0}(T = t_o).$$

It subtracts half of the probability of the observed statistic from the usual exact P-value. The mid P-value has the appealing property that its null expected value for a discrete distribution equals exactly $\frac{1}{2}$, the expected P-value for a continuous distribution. A disadvantage is that a test based on it is no longer "exact," the actual size possibly exceeding the nominal value.

The mid P-value assigns weight $\frac{1}{2}$ to probabilities of all tables comparable to the observed table in the sense that $T = t_o$. For the modified P-value (2.4), the comparable tables are those with $T = t_o$ and $T' = t'_o$. Thus, we can define a mid P version of the modified P-value by

$$P^*_{\text{mid}} = P^* - \frac{1}{2}P_{H_0}(T = t_o,\, T' = t'_o). \tag{2.8}$$

Like the ordinary mid P-value, the modified mid P-value has null expected value equal to $\frac{1}{2}$. The result is easily obtained by noting that the modified mid P-value is a special case of the usual mid P-value using a more refined partitioning of T and T'. The ordinary mid P-value uses a partitioning based on T, and it is the sum of half of $P_{H_0}(T = t_o)$ and the probability of more extreme values of T. The modified

mid P-value uses a partitioning based on T and T' within T. We assume that T and T' have positive values. Let Gap(T) denote the minimum difference between two consecutive values of T. Define a new statistic $T^* = T \times \text{Max}(T')/\text{Gap}(T) + T'$. If Min($T'$) equals 0, we transform from T' to $T'+1$ in order to avoid ties in T^*. Then, $T^*(\mathbf{Z}_1) > T^*(\mathbf{Z}_2)$ for all tables $\mathbf{Z}_1, \mathbf{Z}_2$ with $T(\mathbf{Z}_1) > T(\mathbf{Z}_2)$. Let t_o^* denote the value of T^* for the observed table. Note that a partitioning of the sample space using T and T' within T is equivalent to a partitioning of the sample space using T^*. Since there are no ties, ordering tables using T and T' within T is equivalent to ordering tables using T^*. Then, the sum of half of $P_{H_0}(T = t_o,\, T' = t_o')$ and the probability of more extreme values of T' at $T = t_o$ and more extreme values of T is equivalent to the sum of half of $P_{H_0}(T^* = t_o^*)$ and the probability of more extreme values of T^*. That is,

$$
\begin{aligned}
P_{\text{mid}}^* &= P_{H_0}(T > t_o) + P_{H_0}(T = t_o,\, T' > t_o') + (1/2)P_{H_0}(T = t_o,\, T' = t_o') \\
&= P_{H_0}(T^* > t_o^*) + (1/2)P_{H_0}(T^* = t_o^*).
\end{aligned}
$$

Hence, the modified mid P-value is a special case of the mid P-value with a more refined partitioning, and its null expected value is equal to $\frac{1}{2}$. Also, the difference between the modified P-value and modified mid P-value is less than the difference between the ordinary P-value and ordinary mid P-value. That is, $(P^* - P_{\text{mid}}^*) \le (P - P_{\text{mid}})$.

2.2.3 Examples

We consider the test of conditional independence in three-way contingency tables under the assumption of no three-factor interaction. We will illustrate the ordinary and modified P-values using $2 \times 2 \times 5$ and $2 \times 2 \times 18$ contingency tables. For $2 \times 2 \times K$ tables, the exact test utilizes the test statistic $T = \sum_k n_{11k}$, given

$\{n_{1+k}, n_{2+k}, n_{+1k}, n_{+2k}\}$. It assumes homogeneity of the odds ratios in the $2 \times 2 \times K$ contingency tables. For modified P-values, we can utilize $\sum X_k^2$ or the table probability, $P(\mathbf{Z})$, for the secondary statistic T'. In the examples we utilize $\sum X_k^2$ for T' in (2.4).

We illustrate the modified P-values (2.4) and (2.5) using Table 2.1, taken from Mantel (1963). It refers to the effectiveness of immediately injected or $1\frac{1}{2}$-hour-delayed penicillin in protecting rabbits against lethal injection with β-hemolytic streptococci. Let P=penicillin level, D=delay, and C=whether cured. Under the assumption of a constant odds ratio θ between D and C at each level of P, we test $H_0 : \theta = 1$ against $H_a : \theta > 1$. Our alternative is the higher cure rate for immediate injection. For the first and last table, the zero marginal count implies that the conditional distribution of n_{11k} is degenerate, and the table makes no contribution to the test. Therefore, we can conduct the test using the three remaining tables.

The test statistic is $T = \sum n_{11k}$, given marginal totals of row and column variables at each level of the third one. For these tables, $t_o = 14$, and the four tables with $T \geq 14$ are $\{(n_{111}, n_{112}, n_{113}) = (3,6,6), (2,6,6), (3,5,6), (3,6,5)\}$. The values of T' for these four tables are 11.09, 7.54, 6.59, and 11.09, respectively. Among them, the observed table is $(3,6,5)$. The ordinary exact P-value is $P = P_{H_0}(T \geq 14) = (2+9+16+2)/1452 = 0.0200$. The modified exact P-values are $P^* = P_p^* = (2+2)/1452 = 0.0028$, the null probability for the tables $\{(3,6,6), (3,6,5)\}$.

For another example, we consider Table 2.2, the "crying babies" data given by Cox (1970, p. 5), a $2 \times 2 \times 18$ table. On each of 18 days, babies not crying at a specific time in a hospital ward served as subjects. On each day one baby chosen at random formed the experimental group, and the remainder were controls. Babies were identified as crying or not at the end of a specific period. For these tables, the observed values are t_o=15, t_o'=17.2601 and the P-values are $P = 0.045$, $P^* = 0.024$, and $P_p^* = 0.021$.

There can be a considerable discrepancy between the behavior of the ordinary and modified "exact" P-values, the modified one having a distribution that can be much less discrete. For Table 2.1, the total number of possible P-values equals 9 for the ordinary P-value, 32 for P^*, and 35 for P_p^*. For Table 2.2, the corresponding numbers are 19, 115938, and 13110. Figure 2.1 presents the cumulative distribution functions of the ordinary exact P-value and of P^* for null conditional distributions based on the fixed margins of Table 2.1. Figure 2.2 presents the analogous distributions for P_p^*. Also, Figures 2.3 and 2.4 display the corresponding cumulative distribution functions for null conditional distributions based on the fixed margins of Table 2.2. For Table 2.2, the modified *cdf* for P^* or P_p^* has a distribution practically indistinguishable from the uniform.

We can summarize the degree of conservativeness of each P-value using E_{H_0}(P-value). Using the conditional distribution based on the fixed margins of Table 2.1, $E_{H_0}P =$ 0.611 and $E_{H_0}P^* = 0.545$ and $E_{H_0}P_p^* = 0.542$. For Table 2.2, $E_{H_0}P = 0.576$ and $E_{H_0}P^*$ =0.500 and $E_{H_0}P_p^* = 0.501$.

We now illustrate the ordinary and modified mid P-values. For the modified mid P-value, we can use $T' = \sum X_k^2$ or the table probability for the secondary statistic. For Table 2.1, $P_{\text{mid}} = 0.011$ and $P_{\text{mid}}^* = 0.002$ for both modified mid P-values using $\sum X_k^2$ or the table probability. For Table 2.2, $P_{\text{mid}} = 0.028$, and $P_{\text{mid}}^* = 0.024$ with $T' = \sum X_k^2$ and 0.021 with the table probability. Figures 2.5 and 2.6 present the cumulative distribution functions of the modified exact P-value and the modified mid P-value using $T' = \sum X_k^2$, and the corresponding cumulative distribution functions using the table probability for T', respectively, for null conditional distributions based on the margins of Table 2.1. There is a good contrast between the behavior of the modified "exact" P-value and modified mid P-value. The modified P-value never exceeds the nominal level, but the modified mid P-value can exceed it. The modified

mid P-value jumps and exceeds the nominal value before the modified P-value jumps closely to the nominal value.

Figures 2.7 and 2.8 display the cumulative distribution functions of the ordinary mid P-value and the modified mid P-value using $T' = \sum X_k^2$, and the corresponding cumulative distribution functions using the table probability for the modified mid P-value, respectively, for the null conditional distribution based on the margins of Table 2.1. Though tests based on the ordinary and modified mid P-value are not "exact," the gap between the actual size and the nominal level tends to be less for the modified mid P-value than for the ordinary mid P-value. One way to measure how close the *cdf* of P is to the uniform *cdf* is by the measure

$$M = \int |F(x) - G(x)| dx,$$

where F = *cdf* of P and G = uniform *cdf*. Using Table 2.1 with $T' = \sum X_k^2$, we have $M = 0.055$ for P_{mid}, and $M = 0.022$ for P^*_{mid}. For the exact P-values, we have $M = 0.111$ for P, and $M = 0.045$ for P^*.

Table 2.1. Example for exact analyses.

Penicillin	Response		
Level	Delay	Cured	Died
1/8	None	0	6
	1 1/2 Hour	0	5
1/4	None	3	3
	1 1/2 Hour	0	6
1/2	None	6	0
	1 1/2 Hour	2	4
1	None	5	1
	1 1/2 Hour	6	0
4	None	2	0
	1 1/2 Hour	5	0

Source: Mantel (1963)

Table 2.2. Example for exact analyses.

	Treated		Control	
Day	Not Crying	Crying	Not Crying	Crying
1	1	0	3	5
2	1	0	2	4
3	1	0	1	4
4	0	1	1	5
5	1	0	4	1
6	1	0	4	5
7	1	0	5	3
8	1	0	4	4
9	1	0	3	2
10	0	1	8	1
11	1	0	5	1
12	1	0	8	1
13	1	0	5	3
14	1	0	4	1
15	1	0	4	2
16	1	0	7	1
17	0	1	4	2
18	1	0	5	3

Source: Cox (1970)

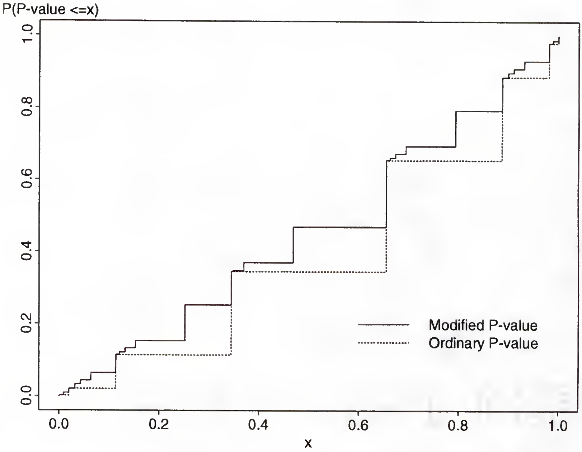

Figure 2.1. Two cumulative distribution functions of exact P-values with $T' = \sum X_k^2$, for the margins of Table 2.1.

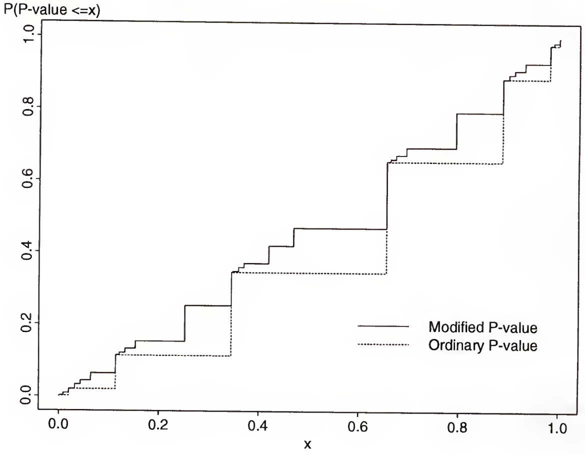

Figure 2.2. Two cumulative distribution functions of exact P-values with $T' = P(\mathbf{Z})$, for the margins of Table 2.1.

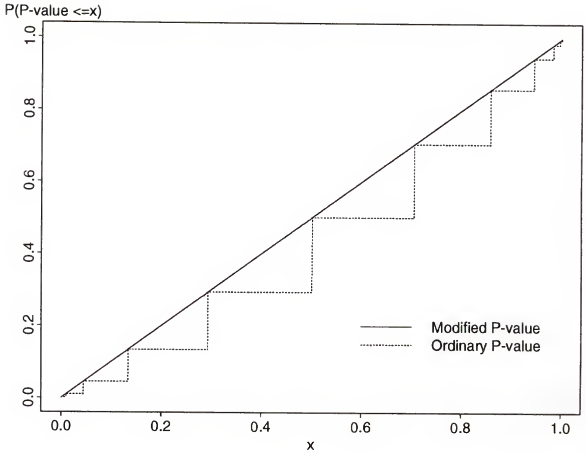

Figure 2.3. Two cumulative distribution functions of exact P-values with $T' = \sum X_k^2$, for the margins of Table 2.2.

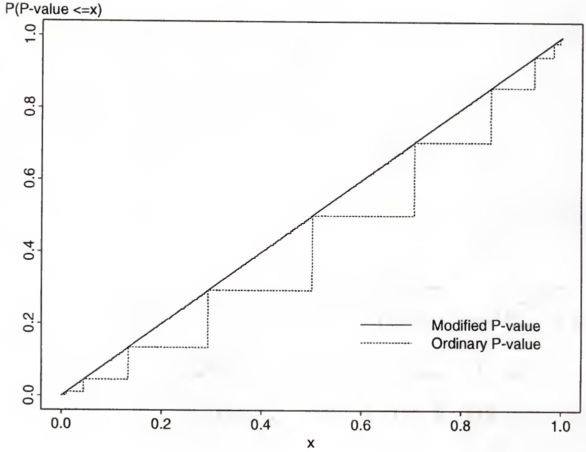

Figure 2.4. Two cumulative distribution functions of exact P-values with $T' = P(\mathbf{Z})$, for the margins of Table 2.2.

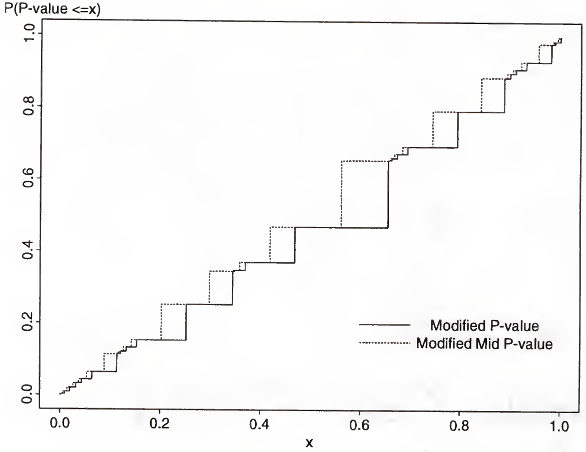

Figure 2.5. Cumulative distribution functions of the modified exact P-value and the modified mid P-value with $T' = \sum X_k^2$, for the margins of Table 2.1.

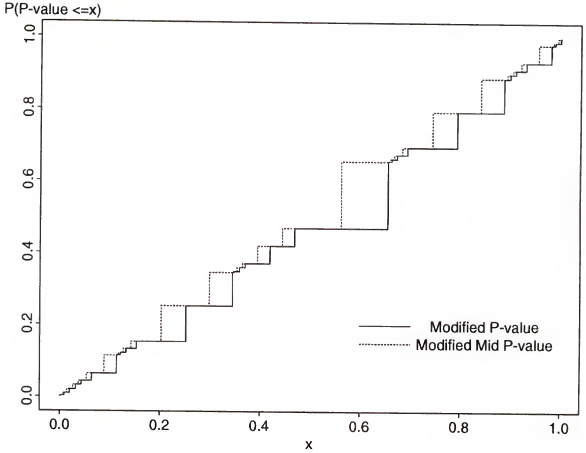

Figure 2.6. Cumulative distribution functions of the modified exact P-value and the modified mid P-value with $T' = P(\mathbf{Z})$, for the margins of Table 2.1.

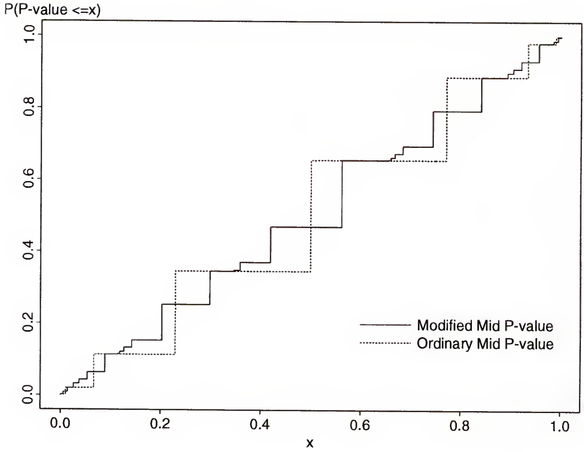

Figure 2.7. Cumulative distribution functions of the ordinary mid P-value and the modified mid P-value with $T' = \sum X_k^2$, for the margins of Table 2.1.

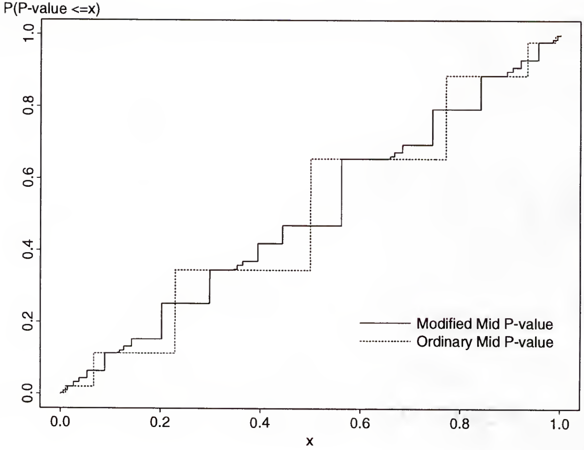

Figure 2.8. Cumulative distribution functions of the ordinary mid P-value and the modified mid P-value with $T' = P(\mathbf{Z})$, for the margins of Table 2.1.

2.2.4 Software

Thomas (1975) gave the first algorithm for exact analysis of several 2×2 contingency tables. This FORTRAN program required enumeration of all possible tables in the conditional reference set; hence, it could be slow. It provided exact tests for conditional independence as well as an exact confidence interval for a common odds ratio, and computed the conditional maximum likelihood estimate. Vollset and Hirji (1991) presented a fast FORTRAN program for the exact test of conditional independence and confidence interval for a common odds ratio in several 2×2 contingency tables.

We suggest modifications of exact methods based on ordering the tables by their secondary statistic. In order to implement a modified exact test, we need to compare the secondary statistic, T', of the generated table to that of the observed table, for tables such that $T = t_o$, and decide whether the table contributes to the P-values. We have modified Vollset and Hirji's FORTRAN program to implement modified exact P-values. Also, the modified software can compute the expected value and the cumulative distribution of P in both ordinary and modified procedures. The source code is listed as Appendix A.

2.3 A Less Conservative "Exact" Confidence Interval

Discreteness also affects confidence interval estimation. For the "exact" confidence interval with nominal confidence coefficient $1 - \alpha$, the actual confidence coefficient is *at least* that level and is unknown (Neyman 1935). Since the modified P-value is less discrete than the ordinary P-value and leads to less conservative "exact" tests, we can

reduce the conservativeness by employing the modified P-value for the construction of confidence intervals.

For $2 \times 2 \times K$ tables, we suggest modified "exact" confidence intervals for an assumed common odds ratio based on inverting results of tests using the modified P-value. Such intervals have confidence coefficient guaranteed to equal at least the nominal level, but are narrower than the ordinary "exact" interval. Illustrations are given for estimating an assumed common odds ratio for several 2×2 tables.

2.3.1 The Ordinary "Exact" Confidence Interval

One can construct an exact confidence interval for a parameter by inverting the exact conditional test regarding the value of that parameter. For an ordinary exact confidence interval, one can invert the test based on the ordinary exact P-value.

To illustrate, suppose we want to estimate an assumed common odds ratio, θ, in a $2 \times 2 \times K$ contingency table. The conditional probability of any table in the reference set, Γ, is

$$P(\{n_{11k}\}|\{n_{1+k}\},\{n_{+1k}\},\{n_{+2k}\};\theta) = \frac{\Pi_k \binom{n_{+1k}}{n_{11k}} \binom{n_{+2k}}{n_{1+k}-n_{11k}} \theta^{n_{11k}}}{\sum_{\mathbf{Z}\in\Gamma} \Pi_k \binom{n_{+1k}}{z_k} \binom{n_{+2k}}{n_{1+k}-z_k} \theta^{z_k}}, \quad (2.9)$$

where $\{z_1, \cdots, z_K\}$ denote values of $\{n_{111}, \cdots, n_{11K}\}$ for a table in the reference set Γ. Let $\Gamma_t = \{\mathbf{Z} : \mathbf{Z} \in \Gamma,\ \sum_k n_{11k} = t\}$. Ordinary exact confidence limits for the common odds ratio are constructed from the conditional distribution of $T = \sum_k n_{11k}$, that is

$$P(T = t;\theta) = \frac{c_t\theta^t}{\sum_{u=t_{\min}}^{t_{\max}} c_u\theta^u}, \quad (2.10)$$

where

$$c_t = \sum_{\mathbf{z}\in\Gamma_t} \Pi_k \binom{n_{+1k}}{z_k} \binom{n_{+2k}}{n_{1+k} - z_k},$$

and where $t_{\min} = \sum_k \max(0, n_{1+k} - n_{+2k})$ and $t_{\max} = \sum_k \min(n_{1+k}, n_{+1k})$. The ordinary interval (Cox 1970, Gart 1970, Mehta *et al.* 1985, Vollset *et al.* 1991) is based on inverting two separate one-sided tests. It equals (θ_-, θ_+), where for $t_{\min} \le t_o \le t_{\max}$,

$$\begin{aligned} \text{at } \theta = \theta_- : \; P_1(\theta) &= \Sigma_{t \ge t_o} P(t;\theta) = \frac{\alpha}{2}, \\ \text{at } \theta = \theta_+ : \; P_2(\theta) &= \Sigma_{t \le t_o} P(t;\theta) = \frac{\alpha}{2}. \end{aligned} \tag{2.11}$$

When $t_o = t_{\min}$, the lower endpoint is 0; if $t_o = t_{\max}$, the upper endpoint is ∞. It is easily shown that $(\theta_-(t), \theta_+(t))$ has confidence coefficient at least $100(1-\alpha)$ (Mehta *et al.* 1985). Due to discreteness of the distribution of T, we have only a conservative confidence interval, and the actual confidence coefficient is unknown.

2.3.2 The Modified "Exact" Confidence Interval

To ensure that the actual confidence coefficient is closer to the nominal value and to obtain a narrower "exact" interval, one can invert the two one-sided tests based on the modified exact P-value. We illustrate this using a secondary statistic $\sum X_k^2(\theta)$ or the table probability to generate the secondary partitioning. In the non-null case, T' is defined as

$$T' = \sum X_k^2(\theta) = \sum_k \sum_i \sum_j \frac{(n_{ijk} - \hat{m}_{ijk}(\theta))^2}{\hat{m}_{ijk}(\theta)},$$

where $\hat{m}_{ijk}(\theta)$ is the estimate of the expected cell count, assuming common odds ratio θ. When $\theta = 1$, $\sum X_k^2(\theta)$ is the Pearson statistic for testing conditional independence.

If large values of T' contradict the null, we let $B(\theta) = \{\mathbf{Z} : \mathbf{Z} \in \Gamma, T = t_o, T'(\theta) \geq t'_o(\theta)\}$. When the table probability is utilized, we denote $P(\mathbf{Z};\theta)$ as the probability of table $\mathbf{Z}$ when the common odds ratio is θ, and let $B(\theta) = \{\mathbf{Z} : \mathbf{Z} \in \Gamma, T = t_o, P(\mathbf{Z};\theta) \leq P(\mathbf{N};\theta)\}$. The modified "exact" confidence limits are found using the functions

$$\begin{aligned} P_1^*(\theta) &= \Sigma_{t>t_o} P(t;\theta) + P[B(\theta);\theta], \\ P_2^*(\theta) &= \Sigma_{t<t_o} P(t;\theta) + P[B(\theta);\theta]. \end{aligned} \tag{2.12}$$

The lower limit, θ_-^*, is the smallest of all θ's to satisfy $P_1^*(\theta) \geq \frac{\alpha}{2}$, and the upper limit, θ_+^*, is the largest of all θ's to satisfy $P_2^*(\theta) \geq \frac{\alpha}{2}$. When $P_1^*(\theta)$ and $P_2^*(\theta)$ are strictly monotone functions of θ, the limits satisfy $P_1^*(\theta_-^*) = P_2^*(\theta_+^*) = \frac{\alpha}{2}$.

We show that the probability that this interval excludes θ, $\Pr(\theta_-^* > \theta) + \Pr(\theta_+^* < \theta)$, is at most α. The lower limit is the smallest value of θ for which $P_1^*(\theta) \geq \frac{\alpha}{2}$. For $\theta < \theta_-^*$, $P_1^*(\theta) < \frac{\alpha}{2}$. It follows that

$$\begin{aligned} \Pr(\theta_-^* > \theta) &\leq \Pr(P_1^*(\theta) < \frac{\alpha}{2}) \\ &\leq \Pr(P_1^*(\theta) \leq \frac{\alpha}{2}) \\ &= E \Pr(P_1^*(\theta) \leq \frac{\alpha}{2} | \boldsymbol{m}) \\ &\leq \frac{\alpha}{2}, \end{aligned}$$

where $\boldsymbol{m}$ denotes a possible marginal configuration, and the last step follows because of discreteness. For the upper limit, by the same arguments we have $\Pr(\theta_+^* < \theta) \leq \frac{\alpha}{2}$. The result follows.

Clearly, this interval is contained within the ordinary one. Hence, the modified confidence interval is "exact," yet it has actual confidence coefficient closer to the nominal value than the ordinary "exact" interval. One can solve for the modified

endpoints numerically, based on the ordinary endpoints as the initial values. The algorithm to find the endpoints is as follows. Start with an initial value based on the ordinary one, since the modified limits are contained within the ordinary ones. Note that $P_1(\theta)$ and $P_2(\theta)$ are strictly monotone functions of θ (Mehta *et al.* 1985). Also note that $P_1^*(\theta)$ is bounded by $P_1(\theta)$, and $P_2^*(\theta)$ is bounded by $P_2(\theta)$. Even though $P_1^*(\theta)$ and $P_2^*(\theta)$ are not monotone functions of θ, the limits can be found within the ordinary limits because they are bounded by $P_1(\theta)$ and $P_2(\theta)$, respectively. Hence ordinary confidence limits provide good starting values for both the monotone case and the non-monotone case. The initial value for the lower limit can be set to be θ_-, and the initial value for the upper limit can be set to be $1.01 \times \theta_+$.

Suppose we want to find the lower limit. Generally, the searching algorithm is composed of two steps. The first step is to increase the value of θ until some value of θ has $P_1^*(\theta) \geq \frac{\alpha}{2}$. For the sake of the non-monotone case, the value of θ is increased by a small amount so that $P_1^*(\theta)$ can not change much between two values of θ's. The second step is iteration within an interval to find the limit. Denote by θ_A the most recent estimate that has $P_1^*(\theta) < \frac{\alpha}{2}$ and denote by θ_B the most recent estimate that has $P_1^*(\theta) \geq \frac{\alpha}{2}$. The initial values of θ_A and θ_B are set to be zero. As θ changes, θ_A or θ_B is updated depending on the value of $P_1^*(\theta)$, and these values will be used for the second stage to determine an interval for iteration.

More specifically, if $P_1^*(\theta) < \frac{\alpha}{2}$, the current estimate is too small. If $P_1^*(\theta) > \frac{\alpha}{2}$, the current estimate is too large. For the first step, compute $P_1^*(\theta)$ at the initial value of θ. If $P_1^*(\theta) = \frac{\alpha}{2}$, this is the limit. If $P_1^*(\theta) < \frac{\alpha}{2}$, multiply θ by 1.01 to increase the value of θ. Using this new estimate, compute $P_1^*(\theta)$. Continue this process until some estimate is found that has $P_1^*(\theta) \geq \frac{\alpha}{2}$. Once this happens, the second step begins. Iteration occurs between two values of θ. These two values are the previous estimate that has $P_1^*(\theta) < \frac{\alpha}{2}$ and the current estimate that has $P_1^*(\theta) \geq \frac{\alpha}{2}$. Note that θ_A and θ_B have been updated as the estimate changes. Then the new estimate is defined as

$\frac{\theta_A+\theta_B}{2}$, and $P_1^*(\theta)$ is computed using this estimate. Depending on the value of $P_1^*(\theta)$, θ_A or θ_B is updated. The process continues until $\frac{\theta_A}{\theta_B}-1$ is sufficiently close to zero, for example, 10^{-14}. If $P_1^*(\theta)$ and $P_2^*(\theta)$ are strictly monotone functions, this algorithm finds the limits that satisfy $P_1^*(\theta_-^*) = P_2^*(\theta_+^*) = \frac{\alpha}{2}$. If not a monotone function, it finds the smallest of all θ's to satisfy $P_1^*(\theta) \geq \frac{\alpha}{2}$, and the largest of all θ's to satisfy $P_2^*(\theta) \geq \frac{\alpha}{2}$. Thus, this algorithm can be used for both monotone and non-monotone cases.

For the upper limit, the same procedure follows except that at $\theta = \theta_+$ if $P_1^*(\theta) < \frac{\alpha}{2}$, multiply θ by 0.99 to decrease the value of θ. This comes from the fact that if $P_1^*(\theta) < \frac{\alpha}{2}$, the current estimate is too large, and if $P_1^*(\theta) > \frac{\alpha}{2}$, the current estimate is too small. This algorithm is an adaptation of one written by Baptista and Pike (1977) for exact two-sided confidence limits for an odds ratio in a 2×2 table.

Next, we show that when the ordinary P-value and the modified P-value P_p^* based on table probabilities are identical, then the ordinary and modified exact confidence intervals (based on inverting the test using P_p^*) also are identical. Suppose we use the table probability for T'. By the definition,

$$\begin{aligned} P &= P_{H_0}(T \geq t_o), \\ P_p^* &= P_{H_O}(t > t_o) + P_{H_0}(\{\mathbf{Z} : T = t_o, P(\mathbf{Z}) \leq P(\mathbf{N})\}). \end{aligned}$$

When the ordinary and modified P-values are identical, we have

$$P_{H_0}(T = t_o) = P_{H_0}(\{\mathbf{Z} : T = t_o, P(\mathbf{Z}) \leq P(\mathbf{N})\}).$$

Hence, the observed table has the largest null probability among those tables having $T = t_o$. This means that when $\theta = 1$, the coefficient for the observed table

$$\Pi_k \begin{pmatrix} n_{+1k} \\ n_{11k} \end{pmatrix} \begin{pmatrix} n_{+2k} \\ n_{1+k} - n_{11k} \end{pmatrix}$$

is the largest among those coefficients for tables having $T = t_o$. Since for arbitrary θ we get

$$\Pi_k \begin{pmatrix} n_{+1k} \\ n_{11k} \end{pmatrix} \begin{pmatrix} n_{+2k} \\ n_{1+k} - n_{11k} \end{pmatrix} \theta^{n_{11k}} = [\Pi_k \begin{pmatrix} n_{+1k} \\ n_{11k} \end{pmatrix} \begin{pmatrix} n_{+2k} \\ n_{1+k} - n_{11k} \end{pmatrix}] \theta^{t_o},$$

the table probability for arbitrary θ depends on only this coefficient. Because the observed table has the largest coefficient among those tables having $T = t_o$, it has the largest probability among those tables having $T = t_o$ for arbitrary θ. Hence, $P(T = t_o; \theta) = P[B(\theta); \theta]$, and the ordinary and modified exact confidence intervals also are identical.

This property does not hold when $T' = \sum X_k^2(\theta)$ is used to construct the modified P-value. The expected cell counts in T' have explicit forms under the null, but they do not have explicit forms under the alternative assuming θ, though they can be obtained by the iterative proportional fitting algorithm. For those tables having $T = t_o$, if the observed table has the smallest value of T' under the null, it does not necessarily have the smallest value of T' under the alternative. Hence, the ordinary and modified exact confidence intervals are not necessarily identical when $P = P^*$.

We now illustrate exact confidence intervals for a common odds ratio using Tables 2.1 and 2.2. The 95% "exact" interval using the ordinary approach is $(1.08, 531.51)$ for Table 2.1 and $(0.86, 21.37)$ for Table 2.2. The corresponding modified "exact" confidence interval using $T' = \sum X_k^2(\theta)$ is $(2.08, 67.35)$ for Table 2.1 and $(1.01, 13.63)$ for Table 2.2. Also, the corresponding modified "exact" confidence interval using the table probability for T' is $(2.08, 67.35)$ for Table 2.1 and $(1.04, 14.87)$ for Table 2.2. We see that inferences can be considerably sharper with the modified approach. For Table 2.1, for instance, the lower bound of the ordinary interval indicates that the true odds ratio could be quite close to conditional independence. The modified interval suggests that the odds ratio is substantively quite different from conditional independence.

2.4 Alternative Modifications of "Exact" Confidence Intervals

In previous sections, we have considered two types of probabilities, that is, the probability of obtaining T equal to or less than the observed value of $T = t_o$, and separately the probability of obtaining T equal to or greater than the observed value t_o. Then, confidence limits are constructed by inverting the test. Hence, confidence intervals discussed so far are based on inverting two separate one-sided tests of level $\alpha/2$ each. We now suggest an alternative way to form an "exact" confidence interval for a common odds ratio. This method is based on inverting a single two-sided test rather than two one-sided tests.

We show that confidence intervals based on inverting two-sided tests tend to be less conservative than those based on inverting two separate one-sided tests. Also we discuss modified mid P confidence intervals based on inverting one-sided or two-sided tests using modified mid P-values.

2.4.1 The Ordinary Two-Sided "Exact" Confidence Interval

Sterne (1954) used a two-sided approach in constructing a confidence interval for a single binomial parameter, and Baptista and Pike (1977) used it to construct confidence limits for the odds ratio in a 2×2 table. We can extend this directly to $2 \times 2 \times K$ tables. For testing a particular value of θ, a two-sided P-value is given by

$$P(\theta) = \sum_{\{t \,:\, P(t;\theta) \leq P(t_o;\theta)\}} P(t;\theta). \qquad (2.13)$$

When the distribution of T has probabilities monotonically increasing in t up to some point and then monotonically decreasing after that, this is simply a two-tail

probability. (This has happened for all examples we have considered, and it may indeed be a property of the distribution of T for $2 \times 2 \times K$ tables; however, except for $K = 1$, it does not seem to be known whether the distribution of a sum of noncentral hypergeometric variates is unimodal.) The two-sided exact confidence interval then consists of the values for θ for which this two-sided P-value equals at least α. Alternatively, one could base the two-sided P-value on a non-null test statistic (such as the score statistic), and construct the confidence interval by inverting that test using the exact non-null distribution. We will discuss this in Chapter 5.

This two-sided approach produces an interval that is usually, but not necessarily, shorter than the ordinary one based on inverting two separate one-sided tests. Under certain conditions, it can be shown that the two-sided approach is better, at least for one of the endpoints. For instance, when the upper limit θ_+ of this interval is quite large, the distribution of T often satisfies $P(t;\theta_+) > P(t_o;\theta_+)$ for all $t > t_o$. A special case of this holds when the probabilities are monotone increasing in t, which is guaranteed when $\theta_+ > \max_t\{c_{t-1}/c_t\}$. In order to show this, from (2.10) we have

$$P(T = t;\theta_+) = \frac{c_t\theta_+^t}{\sum_{u=t_{\min}}^{t_{\max}} c_u\theta_+^u}.$$

For $t_{\min} < t \leq t_{\max}$,

$$\begin{aligned} P(T = t;\theta_+) - P(T = t-1;\theta_+) &= \frac{1}{\sum_{u=t_{\min}}^{t_{\max}} c_u\theta_+^u}(c_t\theta_+^t - c_{t-1}\theta_+^{t-1}) \\ &= \frac{\theta_+^{t-1}}{\sum_{u=t_{\min}}^{t_{\max}} c_u\theta_+^u}(c_t\theta_+ - c_{t-1}). \end{aligned}$$

If $\theta_+ > \frac{c_{t-1}}{c_t}$, $P(T = t;\theta_+) > P(T = t-1;\theta_+)$ for arbitrary t. Hence, if $\theta_+ > \max_t\{c_{t-1}/c_t\}$, the probabilities are monotone increasing in t. In this case, since $P(t;\theta_+) > P(t_o;\theta_+)$ for all $t > t_o$,

$$\sum_{\{t\,:\,P(t;\theta_+)\leq P(t_o;\theta_+)\}} P(t;\theta_+) = \sum_{t\leq t_o} P(t;\theta_+) = \alpha.$$

Hence, this upper limit θ_+ is the same as the upper limit obtained using the one-sided testing approach with double the error probability. For instance, the upper limit of the 95% interval based on inverting a two-sided test is then the same as the upper limit of the 90% interval for the approach based on inverting two separate one-sided tests. Analogous remarks apply to the lower limit. In such cases, there is a clear advantage to using this approach based on two-sided tests. Unless one is specifically interested in a one-sided confidence interval (*i.e.*, a lower bound alone or an upper bound alone for θ), we prefer this approach.

2.4.2 The Modified Two-Sided "Exact" Confidence Interval

Following the modified approach of the previous section, one can construct a modification of this confidence interval based on two-sided tests by using a modified P-value. We define a modified two-sided P-value for testing a particular value of θ as

$$P^*(\theta) = P(\theta) - P(\{\mathbf{Z} : \mathbf{Z} \in \Gamma, P(t;\theta) = P(t_o,\theta), T'(\theta) < t'_o(\theta)\}). \tag{2.14}$$

Again, if we use the table probability for the secondary partitioning, we define a modified two-sided P-value for testing a particular value of θ as

$$P^*_p(\theta) = P(\theta) - P(\{\mathbf{Z} : \mathbf{Z} \in \Gamma, P(t;\theta) = P(t_o,\theta), P(\mathbf{Z};\theta) > P(\mathbf{N};\theta)\}). \tag{2.15}$$

For the modified two-sided confidence interval, we consider the shortest interval that contains all of the the values of θ for which

$$P^*(\theta) \geq \alpha. \tag{2.16}$$

The lower limit, θ^*_-, is the smallest θ satisfying (2.16), and the upper limit, θ^*_+, is the largest θ satisfying (2.16). We show that this confidence interval is "exact." For all

values of θ lying outside the closed interval $\theta_{-}^{*} \leq \theta \leq \theta_{+}^{*}$, it follows that $P^{*}(\theta) < \alpha$. Then

$$\begin{aligned}
\Pr(\theta < \theta_{-}^{*}, \theta > \theta_{+}^{*}) &\leq \Pr(P^{*}(\theta) < \alpha) \\
&\leq \Pr(P^{*}(\theta) \leq \alpha) \\
&= E\Pr(P^{*}(\theta) \leq \alpha | \boldsymbol{m}) \\
&\leq \alpha.
\end{aligned}$$

Hence, $\Pr(\theta_{-}^{*} \leq \theta \leq \theta_{+}^{*}) \geq 1 - \alpha$.

This approach gives even narrower intervals than obtained by inverting the two-sided test with the ordinary P-value. Note that θ_{-} is the smallest θ satisfying $P(\theta) \geq \alpha$. Thus, before θ_{-}, there is no point having $P(\theta) \geq \alpha$. Also note that $P^{*}(\theta)$ is bounded by $P(\theta)$ and $P^{*}(\theta) \leq P(\theta)$. For instance, at the ordinary lower limit, if $P^{*}(\theta_{-}) = P(\theta_{-})$, then $\theta_{-}^{*} = \theta_{-}$. Otherwise, $\theta_{-}^{*} \geq \theta_{-}$. By a symmetric argument, $\theta_{+}^{*} \leq \theta_{+}$. Hence, the two-sided modified confidence interval is contained within the two-sided ordinary confidence interval.

We illustrate these alternative "exact" confidence intervals for the common odds ratio using Tables 2.1 and 2.2. For Table 2.1 the 95% confidence interval by inverting a two-sided test is (1.29, 261.49) based on the ordinary exact P-values and (1.38, 40.45) based on modified exact P-values, $P^{*}(\theta)$ and $P_{p}^{*}(\theta)$. Using Table 2.2 the confidence intervals are (0.88, 15.92) using the ordinary exact P-values, (1.01, 10.30) using $P^{*}(\theta)$, and (1.01, 11.14) using $P_{p}^{*}(\theta)$.

Table 2.3 contains 95% confidence intervals obtained using the two separate one-sided ordinary and modified exact P-values, and using the ordinary and modified two-sided exact P-values. For these tables, the confidence interval constructed using the ordinary two-sided P-value is shorter than the ordinary one based on two one-sided P-values. In fact, for each data set, the upper endpoint for the two-sided based interval equals the endpoint that would be obtained with the one-sided method for

a 90% confidence interval. For each type of interval, the ones based on the modified P-value are narrower yet. For Table 2.2 the modified confidence interval based on $T' = \sum_k X_k^2$ is shorter than the corresponding confidence interval based on the table probability in both one-sided and two-sided cases.

One way to compare the methods to construct the confidence interval and to calculate some degree of the conservativeness is using the coverage function (Vollset and Hirji 1991). The coverage function, for a given value of θ, is computed by summation of $P(t;\theta)$ over t for which the confidence interval contains the given value of θ. The function is then plotted as a function of θ. Hence, it displays how closely the actual coverage probability falls to the nominal coverage probability.

For the conditional distribution having the fixed marginal counts of Table 2.1, Figures 2.9 and 2.10 show the actual coverage probability as a function of the true log odds ratio, for 95% confidence intervals based on inverting separate one-sided tests using the ordinary or modified P-value. We use $\sum X_k^2(\theta)$ for Figure 2.9 and the table probability for Figure 2.10, for the secondary partitioning in the modified P-value. There is a clear advantage to using the interval based on the modified P-value. For Table 2.2, this calculation requires a huge computing time, and we have not been able to get results using the conditional distribution based on the margins of all 18 partial tables. Thus, we display results using various subsets of the partial tables of Table 2.2. Figure 2.11 gives an analogous display using various numbers of partial tables from Table 2.2. It shows how the conservativeness is reduced by using confidence intervals based on inverting tests with modified P-values. As the number of strata increases, the modified approach yields actual level closer to the nominal level, and this holds over a broader range of odds ratio values.

For either approach, for sufficiently large θ, all tables with those margins would have lower bound of the interval below θ; for sufficiently small θ, all tables would have upper bound above θ. In such cases, the actual probability of coverage of a

$100(1-\alpha)\%$ confidence interval has lower bound $1-\alpha/2$. That bound is achieved at values of θ that are potential endpoints of the intervals (Neyman 1935). To show this, let (θ_-, θ_+) denote the ordinary interval based on a one-sided test. Suppose that the value of the upper limit, θ_+, is large enough so that all the lower limits from other possible tables are less than θ_+. Since θ_+ is constructed by inverting the one-sided $\alpha/2$ test, we have $P(T \leq t_o; \theta_+) = \alpha/2$ and $P(T \geq t_o + 1; \theta_+) = 1-\alpha/2$ accordingly. The coverage function at $\theta = \theta_+$ is

$$\begin{aligned} C(\theta_+) &= \sum_t I(t, \theta_+) P(t; \theta_+) \\ &= P(t;\ t \geq t_o + 1; \theta_+) \\ &= 1 - \alpha/2, \end{aligned}$$

where $I(t, \theta_+)$ is a indicator function to indicate whether or not θ_+ is within the confidence interval at $T = t$. Note that at $\theta = \theta_+$, we have $P(T \leq t_o; \theta_+) = \alpha/2$, and θ_+ is the upper limit. At some value of $T = t'$, the fact that θ_+ is within this interval corresponds to $P(T \leq t'; \theta_+) > \alpha/2$. In order to satisfy this, we need to have $t' \geq t_o + 1$, since $P(T \leq t_o; \theta_+) = \alpha/2$. Hence, the coverage probability that is the summation of $P(t; \theta_+)$ over t such that $t \geq t_o + 1$ is $1-\alpha/2$. For $\theta > \theta_+$ the coverage function has $P(\theta) \geq 1 - \alpha/2$.

Figures 2.12 and 2.13 give an analogous display for the confidence intervals based on inverting two-sided tests using the ordinary or modified P-value using Table 2.1. For the secondary statistic T', Figure 2.12 uses $\sum X_k^2(\theta)$ and Figure 2.13 uses the table probability. Again, there is an advantage to the interval based on the modified P-value. Comparing the figures of coverage probability for confidence intervals, we see there is almost always an advantage to using the confidence interval based on inverting two-sided tests. Figure 2.14 gives an analogous display using some fixed sets of margins of Table 2.2. There is a dramatic improvement in the two-sided modified confidence intervals, when the number of strata is large. As the number of

strata increases, we can expect that actual coverage probability is very close to the nominal coverage probability. When $\log\theta$ is between -2 and 2, we see there is a large increase in the coverage probability for both the ordinary two-sided and modified two-sided confidence intervals. At that point, many new tables for which the confidence intervals contain the given value of θ are added to the calculation of the coverage probability, and the jump comes from the new included non-null table probabilities. For the coverage probability based on two-sided ordinary tests, the big jump has occurred before the coverage probability based on two-sided modified tests has a big jump, and the amount of increase is greater than that of two-sided modified tests. Also, at that jump point, more new tables are included for the coverage probability based on two-sided ordinary tests than the coverage probability based on two-sided modified tests.

We have observed similar results using other sets of fixed margins. In particular, for the two-sided approach, for large $|\log\theta|$, the true coverage probability has 0.95 as a lower bound rather than 0.975. For the proof, let (θ_-, θ_+) be the ordinary confidence interval based on the two-sided test. Suppose that the value of the upper limit, θ_+, is large enough so that all of the lower limits from other possible tables are less than θ_+. Then at $\theta = \theta_+$ we have $\sum_{\{t\,:\,P(t;\theta_+)\leq P(t_o;\theta_+)\}} P(t;\theta_+) \leq \alpha$, accordingly, $\sum_{\{t\,:\,P(t;\theta_+)>P(t_o;\theta_+)\}} P(t;\theta_+) \geq 1-\alpha$. At $\theta = \theta_+$, the coverage function is

$$\begin{aligned} C(\theta_+) &= \sum_t I(t,\theta_+)P(t;\theta_+) \\ &= \sum_{\{t\,:\,P(t;\theta_+)>P(t_o;\theta_+)\}} P(t;\theta_+) \\ &\geq 1-\alpha, \end{aligned}$$

since at $\theta = \theta_+$, we have $\sum_{\{t\,:\,P(t;\theta_+)\leq P(t_o;\theta_+)\}} P(t;\theta_+) \leq \alpha$. At some value of $T = t'$, the fact that θ_+ is within this interval corresponds to $\sum_{\{t\,:\,P(t;\theta_+)\leq P(t';\theta_+)\}} P(t;\theta_+) > \alpha$. In order to satisfy this, we need to have $P(t';\theta_+) > P(t_o;\theta_+)$. Then the two-sided

ordinary P-value is larger than α at $T = t'$. Hence, the coverage probability, which is the summation over t such that $P(t;\theta_+) > P(t_o;\theta_+)$, is at least $1-\alpha$. Also for $\theta > \theta_+$ the coverage function has $P(\theta) \geq 1-\alpha$.

For a special case, suppose that $P(t;\theta_+) > P(t_o;\theta_+)$ for all $t > t_o$. Then at $\theta = \theta_+$,

$$P(\theta_+) = \sum_{\{t\,:\,P(t;\theta_+)\leq P(t_o;\theta_+)\}} P(t;\theta_+) = \sum_{t\leq t_o} P(t;\theta_+) \quad = \quad \alpha.$$

Accordingly, we have $\sum_{t\geq t_o+1} P(t;\theta_+) = 1-\alpha$. Then the coverage function at $\theta = \theta_+$ is

$$\begin{aligned} C(\theta_+) &= \sum_t I(t,\theta_+)P(t;\theta_+) \\ &= \sum_{t\geq t_o+1} P(t;\theta_+) \\ &= 1-\alpha, \end{aligned}$$

since at some value of $T = t'$, the fact that θ_+ is within this interval corresponds to $P(T \leq t';\theta_+) > \alpha$. This requires $t' \geq t_o + 1$, since $P(T \leq t_o;\theta_+) = \alpha$. Hence the coverage function has $C(\theta_+) \geq 1-\alpha$. This relates to the property mentioned previously, by which an interval endpoint for the two-sided approach with error probability α can equal one for the one-sided approach with error probability 2α.

So far, we have used the coverage probability to compare the methods of constructing the confidence interval. An alternative way to compare them is to compute the expected length of confidence intervals for θ or for $\log\theta$. A complication results from infinite endpoints that occur at $T = t_{\max}$ or $T = t_{\min}$. Figure 2.15 displays the expected length of confidence intervals for θ, for four methods, using the margins of Table 2.1. The two-sided modified confidence interval has the smallest expected length, uniformly for all θ. For instance, the expected lengths at $\theta = 1$ are 21.84, 17.22, 13.78, and 11.21 for one-sided ordinary, one-sided modified, two-sided ordinary, and two-sided modified intervals, respectively. For this figure, we arbitrarily set the

upper limit equal to 1000 whenever $T = t_{\max}$. Since the expected length depends on the upper limit at $T = t_{\max}$, that value was chosen to be almost two times the maximum finite upper limit among the four methods. Figure 2.16 presents the analogous expected length of confidence intervals for $\log\theta$, using the margins of Table 2.1. Again, the two-sided modified confidence interval has uniformly the smallest expected length. We use 1.0×10^{-3} for the lower limit of θ at $T = t_{\min}$ and 1000 for the upper limit of θ at $T = t_{\max}$. Figures 2.17 and 2.18 give analogous displays using the margins of table 2.1, comparing the lengths conditional on $T \neq t_{\min}$ or $t_{\max}$. Then, the expected length does not depend on the values of the lower limit at $t_{\min}$ and the upper limit at $T = t_{\max}$. Again, the two-sided modified confidence interval has uniformly the smallest expected length.

2.4.3 The One-Sided Mid P Confidence Interval

For confidence intervals for a common odds ratio based either on inverting two separate one-sided tests or inverting a two-sided test, one can construct even narrower intervals, albeit not "exact" ones, by inverting the tests based on the modified mid P value. The ordinary mid P confidence limits based on inverting two separate one-sided tests are found using the functions

$$\begin{aligned} P_{\text{mid}(1)}(\theta) &= P_1(\theta) - \frac{1}{2}P(t_o;\theta)), \\ P_{\text{mid}(2)}(\theta) &= P_2(\theta) - \frac{1}{2}P(t_o;\theta)). \end{aligned} \tag{2.17}$$

The limits are determined by the same method used for the modified exact confidence interval, using $P_{\text{mid}(1)}(\theta)$ for the lower limit and $P_{\text{mid}(2)}(\theta)$ for the upper limit. Though

approximate, this type of confidence interval based on the ordinary mid P-value has been observed empirically to behave well (Mehta and Walsh 1992).

Following the modified approach based on using a one-sided modified mid P-value, let $B_1(\theta) = \{\mathbf{Z} : \mathbf{Z} \in \Gamma, T = t_o, T'(\theta) = t'_o(\theta)\}$. The modified mid P confidence interval based on inverting two separate one-sided tests uses

$$\begin{aligned} P^*_{\mathrm{mid}(1)}(\theta) &= P^*_1(\theta) - \frac{1}{2}P(B_1(\theta);\theta), \\ P^*_{\mathrm{mid}(2)}(\theta) &= P^*_2(\theta) - \frac{1}{2}P(B_1(\theta);\theta). \end{aligned} \tag{2.18}$$

The limits are chosen by the same method used for the modified exact confidence interval, using $P^*_{\mathrm{mid}(1)}(\theta)$ for the lower limit and $P^*_{\mathrm{mid}(2)}(\theta)$ for the upper limit. This approach tends to give narrower intervals than obtained by inverting the one-sided test with the ordinary mid P-value. We illustrate these confidence intervals for the common odds ratio using Tables 2.1 and 2.2. For Table 2.1, the 95% confidence interval by inverting a one-sided test is (1.34, 266.54) based on the ordinary mid P-values and (2.22, 56.00) based on the modified mid P-values using $\sum X^2_k(\theta)$ or the table probability for T'. Using Table 2.2, the confidence intervals are (0.98, 16.89) using the ordinary mid P-values, (1.01, 13.61) using the modified mid P-values with $\sum X^2_k(\theta)$, and (1.04, 14.85) using the modified mid P-values with the table probability for T' .

2.4.4 The Two-Sided Mid P Confidence Interval

As the two-sided approach tends to give an interval that is usually narrower than the one based on inverting two separate one-sided tests, we can construct a shorter interval using two-sided mid P-values. Though these cannot guarantee achieving at

least the nominal confidence level, one could define mid P versions of the ordinary two-sided and modified two-sided intervals. For testing a particular value of θ, a two-sided mid P-value can be defined as

$$P_{\text{mid}}(\theta) \quad = \quad P(\theta) - \frac{1}{2}P(\{\mathbf{Z} : \mathbf{Z} \in \Gamma, P(t;\theta) = P(t_o;\theta)\}). \qquad (2.19)$$

The limits are determined by the same method used for the two-sided exact confidence interval.

Following the modified approach, one can construct a modified confidence interval based on two-sided tests by using a modified mid P-value. We define a modified two-sided mid P-value for testing a particular value of θ as

$$P^*_{\text{mid}}(\theta) \quad = \quad P^*(\theta) - \frac{1}{2}P(\{\mathbf{Z} : \mathbf{Z} \in \Gamma, P(t;\theta) = P(t_o;\theta), T'(\theta) = t'_o(\theta)\}). \ (2.20)$$

Also, the limits are determined by the same method used for the two-sided exact confidence interval. We illustrate these confidence intervals for the common odds ratio using Tables 2.1 and 2.2. For Table 2.1, the 95% confidence interval by inverting a two-sided test is (1.38, 131.51) based on the ordinary mid P-values and (1.38, 35.51) based on modified mid P-values using $T' = \sum X_k^2(\theta)$. Using Table 2.2, the confidence intervals are (1.01, 12.58) and (1.01, 10.29) using the ordinary and modified mid P-values with $T' = \sum X_k^2(\theta)$, respectively. For these data sets, the confidence interval constructed by using the ordinary two-sided mid P-values is shorter than the ordinary one based on two one-sided mid P-values. For each type of interval, the modified interval is narrower than the ordinary one. Table 2.4 summarizes these 95% confidence intervals using Table 2.1 and Table 2.2.

For the conditional distribution having the fixed marginal counts of Table 2.1, Figure 2.19 shows the actual coverage probability as a function of the true log odds ratio, for the 95% confidence intervals based on inverting separate one-sided tests using the ordinary mid P-value or the modified mid P-value with $T' = \sum X_k^2(\theta)$. The

exact method yields a coverage exceeding the nominal level, whereas the coverage of the mid P-value fluctuates about the nominal level. For either approach, for sufficiently large $|\log \theta|$, the actual probability of coverage of a $100(1-\alpha)\%$ confidence interval is centered about $1-\alpha/2$ and that of the modified mid P-value deviates less from $1-\alpha/2$.

Figure 2.20 gives an analogous display for the confidence intervals based on inverting two-sided tests using the ordinary mid P-value or the modified mid P-value with $T' = \sum X_k^2(\theta)$. There is an advantage to the interval based on the modified P-value. For either approach, the actual probability of coverage of a $100(1-\alpha)\%$ confidence interval is centered about the nominal level, and that of the modified mid P-value is even closer to the nominal level. For intervals using mid P-values, we suggest the use of the confidence interval based on inverting two-sided tests using the modified mid P-value.

Table 2.3. Various 95% confidence intervals for the common odds ratio.

	Method	Data set 1	Data set 2
Exact CI			
	Ordinary 1-sided P	1.08, 531.51	0.86, 21.37
	Modified 1-sided P (P^*)	2.08, 67.35	1.01, 13.63
	Modified 1-sided P (P_p^*)	2.08, 67.35	1.04, 14.87
	Ordinary 2-sided P	1.29, 261.49	0.88, 15.92
	Modified 2-sided P (P^*)	1.38, 40.45	1.01, 10.30
	Modified 2-sided P (P_p^*)	1.38, 40.45	1.01, 11.14
Approximate CI			
	Mantel-Haenszel	1.03, 47.73	0.86, 12.93
	ML	1.28, 128.12	0.99, 17.64

Table 2.4. Various 95% confidence intervals for the common odds ratio using mid P-value.

	Method	Data set 1	Data set 2
Approximate CI			
	Ordinary 1-sided mid P	1.34, 266.54	0.98, 16.89
	Modified 1-sided mid P (P^*)	2.22, 56.00	1.01, 13.61
	Modified 1-sided mid P (P_p^*)	2.22, 56.00	1.04, 14.85
	Ordinary 2-sided mid P	1.38, 131.51	1.01, 12.58
	Modified 2-sided mid P (P^*)	1.38, 35.51	1.01, 10.29

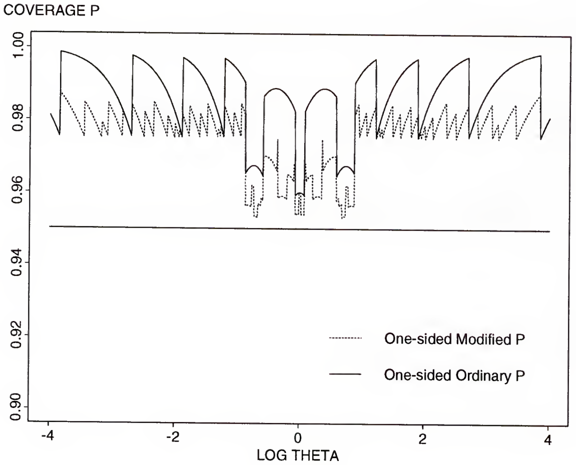

Figure 2.9. Coverage probability for confidence intervals based on inverting one-sided tests with $T' = \sum X_k^2(\theta)$, for conditional distribution based on margins of Table 2.1.

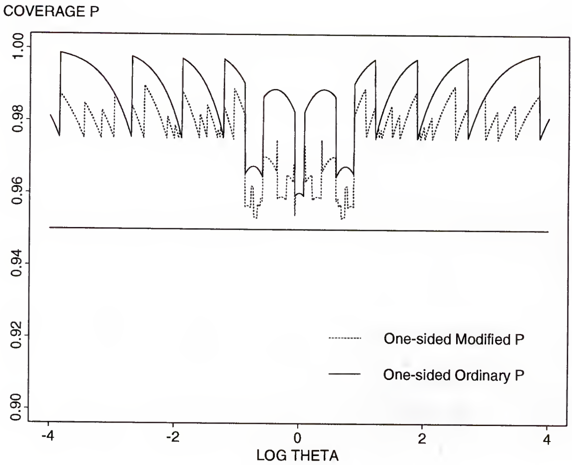

Figure 2.10. Coverage probability for confidence intervals based on inverting one-sided tests with $T' = P(\mathbf{Z})$, for conditional distribution based on margins of Table 2.1.

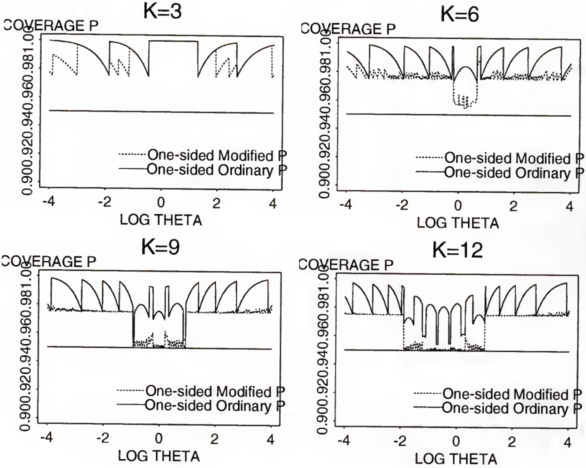

Figure 2.11. Coverage probability for confidence intervals based on inverting one-sided tests with $T' = \sum X_k^2(\theta)$, for conditional distribution based on first K partial tables of Table 2.2.

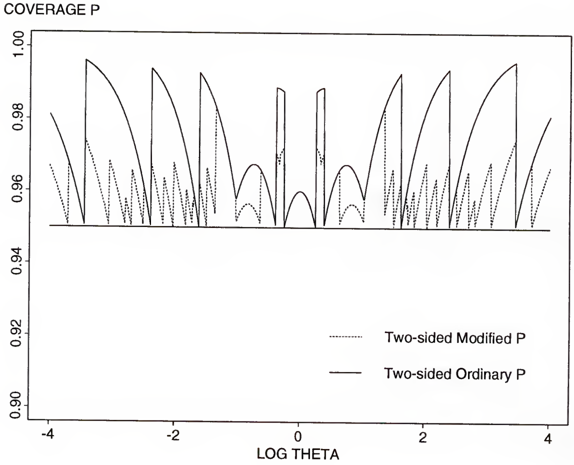

Figure 2.12. Coverage probability for confidence intervals based on inverting two-sided tests with $T' = \sum X_k^2(\theta)$, for conditional distribution based on margins of Table 2.1.

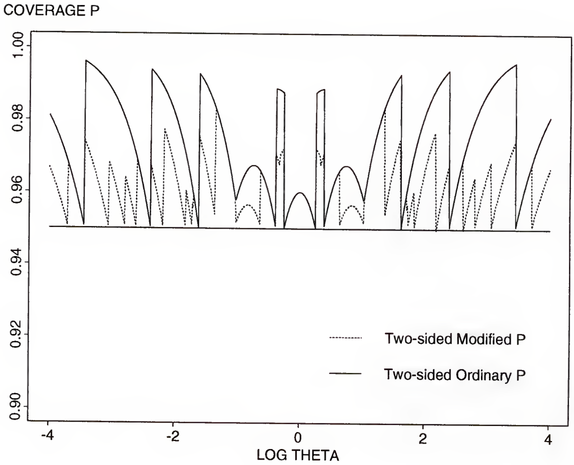

Figure 2.13. Coverage probability for confidence intervals based on inverting two-sided tests with $T' = P(\mathbf{Z})$, for conditional distribution based on margins of Table 2.1.

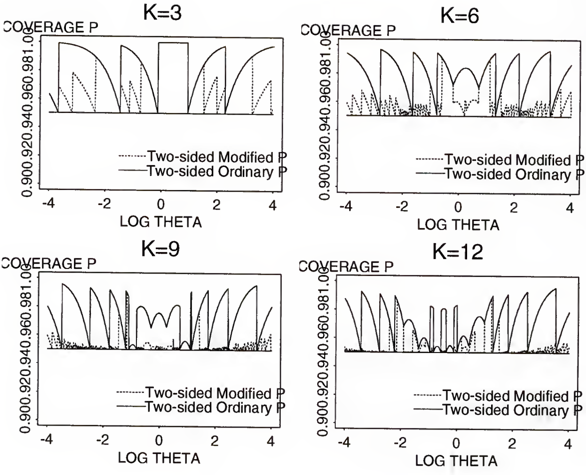

Figure 2.14. Coverage probability for confidence intervals based on inverting two-sided tests with $T' = \sum X_k^2(\theta)$, for conditional distribution based on first K partial tables of Table 2.2.

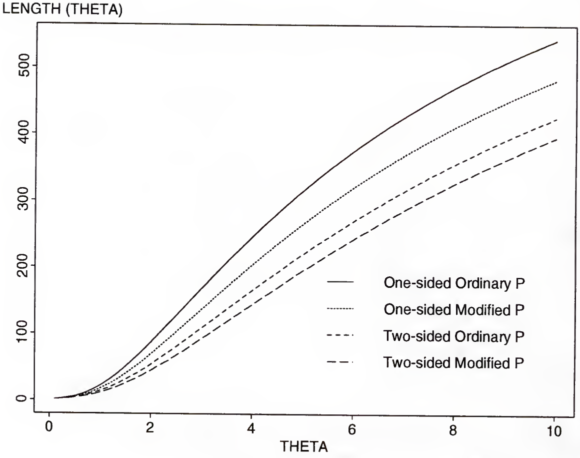

Figure 2.15. Expected length of confidence intervals for θ, with $T' = \sum X_k^2(\theta)$, for conditional distribution based on margins of Table 2.1.

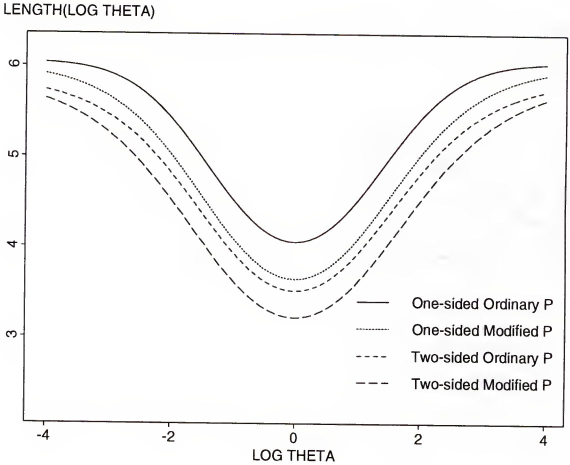

Figure 2.16. Expected length of confidence intervals for log θ, with $T' = \sum X_k^2(\theta)$, for conditional distribution based on margins of Table 2.1.

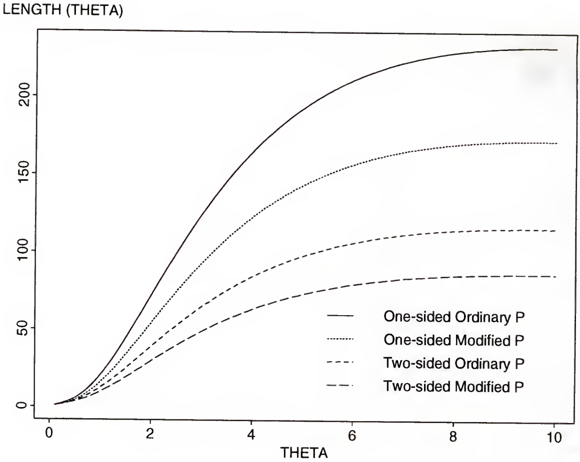

Figure 2.17. Expected length of confidence intervals for θ, conditional on $T \neq t_{\min}$ or $t_{\max}$, with $T' = \sum X_k^2(\theta)$, for conditional distribution based on margins of Table 2.1.

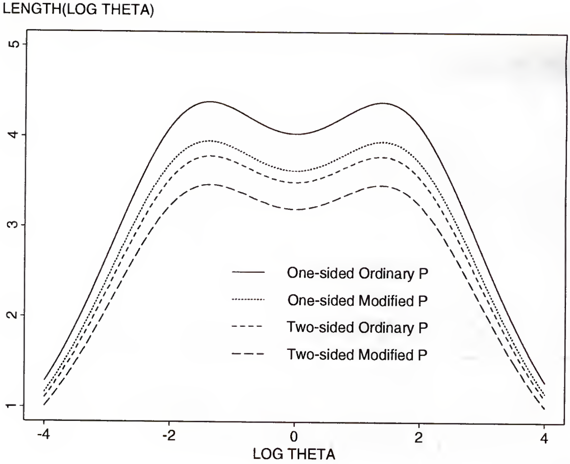

Figure 2.18. Expected length of confidence intervals for log θ, conditional on $T \neq t_{\min}$ or $t_{\max}$, with $T' = \sum X_k^2(\theta)$, for conditional distribution based on margins of Table 2.1.

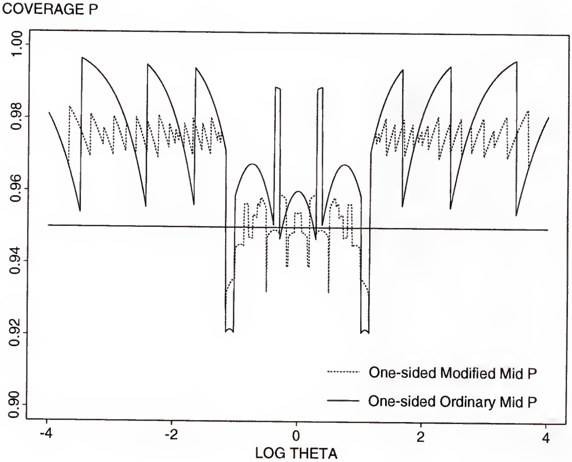

Figure 2.19. Coverage probability for confidence intervals based on inverting one-sided tests using mid P-values with $T' = \sum X_k^2(\theta)$, for conditional distribution based on margins of Table 2.1.

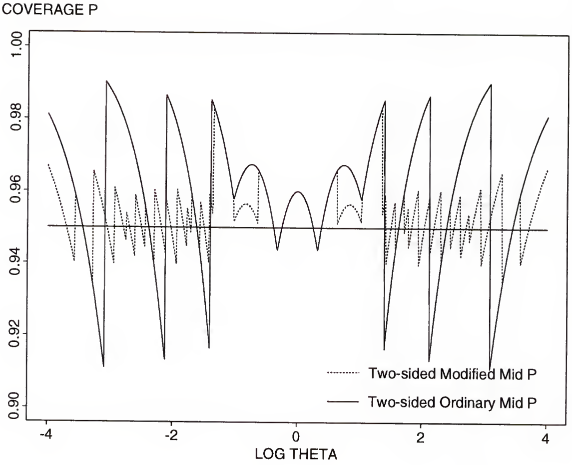

Figure 2.20. Coverage probability for confidence intervals based on inverting two-sided tests using mid P-values with $T' = \sum X_k^2(\theta)$, for conditional distribution based on margins of Table 2.1.

2.5 Connections with Logistic Regression

Consider a set of independent binary variables, $Y_1, \cdots, Y_n$. Corresponding to each variable, Y_j, there is a $(p \times 1)$ vector $\mathbf{x}_j = (x_{1j}, \cdots, x_{pj})'$ of explanatory variables. Let π_j be the probability that $Y_j = 1$. Suppose that the response is related to the explanatory variables by the logistic regression model,

$$\log \frac{\pi_j}{1-\pi_j} = \gamma + \mathbf{x}_j'\boldsymbol{\beta}. \tag{2.21}$$

The likelihood function is

$$Pr(Y_1 = y_1, \cdots, Y_n = y_n) = \frac{\exp[\Sigma_{j=1}^n y_j(\mathbf{x}_j'\boldsymbol{\beta} + \gamma)]}{\Pi_{j=1}^n[1 + \exp(\mathbf{x}_j'\boldsymbol{\beta} + \gamma)]}.$$

The $p \times 1$ vector of sufficient statistic for $\boldsymbol{\beta}$ is $\mathbf{t} = \Sigma_{j=1}^n y_j \mathbf{x}_j$.

Suppose $p = 2$, and we want to conduct inferences about β_1. Again, one can eliminate β_2 by conditioning on its sufficient statistic, $t_2 = \Sigma_j y_j x_{2j}$. One can treat the data for the logistic regression model as a three-way $2 \times I \times K$ tables where I and K are the number of distinct values of the explanatory variables, X_1 and X_2, respectively.

Exact inference in logistic regression often is highly discrete, even degenerate. One can often alleviate this problem somewhat by treating the data as a contingency table and using the alternative way discussed in Section 2 of constructing P-values. To illustrate, for Table 2.1 we let π_{ij} denote the probability of cure for the jth individual at the ith penicillin level. The logistic model has form $\log \frac{\pi_{ij}}{1-\pi_{ij}} = \gamma_i + \beta x_{ij}$ $i = 1, \cdots, 3$, where x_{ij} is a dummy variable for delay. The observed value of the sufficient statistic T is 14. For testing $H_0 : \beta = 0$, the exact one-sided P-value is $P = P(T \geq 14) = 0.0200$. The modified exact P-value, using $T' = \sum X_k^2(\theta)$ or the table probability, is 0.0028.

2.6 Discussion

We have shown that use of a modified P-value leads to exact tests and confidence intervals that are less conservative than the usual ones. The improvement can be considerable when K is large but n is not, in which case there may be a large number of tables with the different secondary statistic value that have the same primary test statistic value.

We prefer modified exact tests and confidence intervals over the ordinary exact ones, because they are less conservative than the ordinary ones but still guarantee at least the nominal level. We prefer confidence intervals based on inverting two-sided tests over those based on inverting two separate one-sided tests, because they tend to be less conservative. Likewise, for confidence intervals using mid P-values, we prefer intervals based on inverting two-sided tests using modified mid P-values.

For the secondary statistic, we have used $\sum_k X_k^2$ and the table probability in our examples, and clearly the reduction in conservativeness occurs with test statistics for more general alternatives. A FORTRAN program has been prepared, designed for IBM-compatible PCs or UNIX workstations, for computing modified P-values for tests of conditional independence and modified confidence intervals for an assumed common odds ratio. This program also computes the actual coverage probability and the expected length of confidence intervals using four methods. This program, for $2 \times 2 \times K$ tables, is an adaptation of one written by Vollset and Hirji (1991) for ordinary exact inference for such tables. The Appendix A contains the FORTRAN source code.

CHAPTER 3
APPROXIMATING EXACT INFERENCE ABOUT CONDITIONAL ASSOCIATION

3.1 Introduction

For three-way tables, consider the hypothesis of conditional independence of X and Y, given Z. This hypothesis is usually tested against the alternative of no three-factor interaction. The general alternative that permits three-factor interaction is the general loglinear model for a three-way table and has the form

$$\log m_{ijk} = \mu + \lambda_i^X + \lambda_j^Y + \lambda_k^Z + \lambda_{ij}^{XY} + \lambda_{ik}^{XZ} + \lambda_{jk}^{YZ} + \lambda_{ijk}^{XYZ}. \tag{3.1}$$

When X or Y are ordinal, narrower alternatives can be constructed for the exact tests.

We suggest exact inference regarding conditional associations in three-way contingency tables. For $I \times J \times K$ tables, we discuss six test statistics for conditional independence that have natural connections with loglinear models for various alternatives. We use a simulation algorithm to obtain precise estimates of exact P-values for cases that are currently computationally infeasible.

For three-way contingency tables, current computational algorithms for the exact methods are restricted to certain analysis for $2 \times J \times K$ tables. Also when the sample size is small or when the contingency tables are sparse, large-sample approximations can be questionable to apply. The Monte Carlo method is an alternative to either the exact or asymptotic methods. This method is based on estimating the exact conditional sampling distribution of the statistic, by generating random tables having the relevant fixed margins. The advantage of this method is that the number of tables

generated is fixed in advance, and the computing time does not depend greatly on the sample size n and the table size, compared to methods for exact analysis. For the random table generation, we use the procedure by Patefield (1981) that simulates hypergeometric distributions.

Section 2 discusses exact tests of conditional independence in $I \times J \times K$ tables using three statistics that are popular for asymptotic tests. These are naturally linked to alternatives corresponding to loglinear models that assume a lack of three-factor interaction. Section 3 presents three other statistics that do not require this assumption. All six test statistics are score statistics for loglinear models that treat none, one, or both of the classifications as ordinal. Section 4 discusses possible alternative ways of forming modified exact P-values in $I \times J \times K$ contingency tables, generalizing the modified P-value discussed in Chapter 2. We propose modified exact P-values for six tests for testing conditional independence with $I \times J \times K$ tables.

Computational algorithms have limited availability for tests of conditional independence when I and J exceed two. Section 5 describes a Monte Carlo sampling routine that approximates the ordinary and modified exact P-values. We utilize six test statistics for exact tests of conditional independence. Section 6 illustrates approximate exact tests of conditional independence with examples, and Section 7 explains a FORTRAN program utilizing the simulation algorithm.

3.2 Tests of Conditional Independence Assuming No Three-factor Interaction

This section presents three test statistics for testing conditional independence of X and Y, given Z, in $I \times J \times K$ contingency tables, proposed by Birch (1965). We present loglinear models for which these are score statistics. These models assume a lack of three-factor interaction. We then present three adaptations of these statistics

that do not require that assumption in the next section. In each case, one test treats both X and Y as nominal, one test treats X as nominal and Y as ordinal, and one test treats both as ordinal.

The asymptotic chi-squared theory is well developed for the statistics we present. Our focus will be to construct exact tests of conditional independence, using these statistics with the reference set Γ of tables with the same margins. We use score statistics for loglinear models rather than likelihood-ratio or Wald statistics. This makes the computations for exact analyses simpler, since one does not need to fit the model for each table in Γ.

3.2.1 Nominal-by-Nominal Test

Birch (1965), Landis *et al.* (1978), and Mantel and Byar (1978) generalized the Cochran-Mantel-Haenszel statistic to handle more than two groups or more than two responses. Suppose X and Y are nominal. Let $\mathbf{n}_k$ denote the counts for cells in the first $I-1$ rows and $J-1$ columns for stratum k of Z. Conditional on the row and column totals in that stratum, let $\mathbf{m}_k$ denote the null expected value of $\mathbf{n}_k$. Then $\mathbf{d} = \Sigma_k(\mathbf{n}_k - \mathbf{m}_k)$ represents the $(I-1)(J-1) \times 1$ vector having elements,

$$d_{ij} = \Sigma_k[n_{ijk} - (\frac{n_{i+k}n_{+jk}}{n_{++k}})], \qquad i = 1, \cdots, I-1 \qquad j = 1, \cdots, J-1. \qquad (3.2)$$

Let $\mathbf{V}_k$ denote the null covariance matrix of $\mathbf{n}_k$, where

$$\mathrm{Cov}(n_{ijk}, n_{i'j'k}) = \frac{n_{i+k}(\delta_{ii'}n_{++k} - n_{i'+k})n_{+jk}(\delta_{jj'}n_{++k} - n_{+j'k})}{n_{++k}^2(n_{++k} - 1)} \qquad (3.3)$$

$$\text{with } \delta_{ab} = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{otherwise.} \end{cases}$$

Then $\mathbf{V} = \Sigma_k \mathbf{V}_k$ is the null covariance matrix of $\mathbf{d}$. The efficient score statistic for testing conditional independence against the alternative of no three-factor interaction is

$$C^2 = \boldsymbol{d}'\boldsymbol{V}^{-1}\boldsymbol{d}. \tag{3.4}$$

This is also called the generalized Cochran-Mantel-Haenszel statistic. Under conditional independence, this statistic has a large sample chi-squared distribution with $df = (I-1)(J-1)$. For $K = 1$ stratum with n observations, the statistic reduces to the multiple $(n-1)/n$ of the Pearson chi-squared statistic for testing independence.

The statistic C^2 is sensitive to detecting conditional associations when the association is similar in each stratum. Hence, the generalized Cochran-Mantel-Haenszel statistic has low power for detecting an association in which the patterns of association for some of the strata are in the opposite direction of the patterns displayed by other strata, relative to the case that the association is similar.

3.2.2 Ordinal-by-Ordinal Test

When X and Y are ordinal, it often makes sense to test against a narrow alternative, corresponding to a monotone trend in the conditional association. It then makes sense to form a test statistic using a model that is a special case of the no three-factor interaction model and reflects the ordinality, such as the model of homogeneous linear-by-linear association,

$$\log m_{ijk} = \mu + \lambda_i^X + \lambda_j^Y + \lambda_k^Z + \beta u_i v_j + \lambda_{ik}^{XZ} + \lambda_{jk}^{YZ}. \tag{3.5}$$

It replaces the general association term λ_{ij}^{XY} by a linear-by-linear term $\beta u_i v_j$, where $\{u_i\}$ and $\{v_j\}$ are monotone scores for levels of X and Y. The parameter β in that model describes $X - Y$ partial association. The model of conditional independence

of X and Y is its special case in which $\beta = 0$. For this model, the sufficient statistic for β is $\Sigma_k[\Sigma_i\Sigma_j u_i v_j n_{ijk}]$. When $I = J = 2$, the usual statistic $\sum n_{11k}$ results from the scores $u_1 = v_1 = 1, u_2 = v_2 = 0$. This is the Birch's exact test statistic for testing conditional independence in $2 \times 2 \times K$ contingency tables, and we have utilized this statistic in Chapter 2 for the conditional exact test. Also, Mehta, Patel and Gray (1985) and Vollset, Hirji and Elashoff (1991) used this statistic to implement the exact test.

For the asymptotic test of $H_o: \beta = 0$, one can use Mantel's (1963) generalized statistic for detecting association between ordinal variables. This ordinal test focuses the departure from independence on a single degree of freedom. Suppose we expect a monotone conditional relationship between X and Y, with the same direction at each level of Z, and suppose that we can assign monotone scores $\{u_i\}$ to levels of X and $\{\nu_j\}$ to levels of Y. Then there is evidence of positive trend if, within each stratum, the statistic $\Sigma_i\Sigma_j u_i v_j n_{ijk}$ is greater than its expectation under independence.

For the model (3.5), given the marginal totals in each stratum and under conditional independence of X and Y,

$$E(\Sigma_i\Sigma_j u_i v_j n_{ijk}) = \frac{(\Sigma_i u_i n_{i+k})(\Sigma_j v_j n_{+jk})}{n_{++k}},$$

$$\mathrm{Var}(\Sigma_i\Sigma_j u_i v_j n_{ijk}) = \frac{1}{n_{++k}-1}[\Sigma_i u_i^2 n_{i+k} - \frac{(\Sigma_i u_i n_{i+k})^2}{n_{++k}}] \times [\Sigma_j v_j^2 n_{+jk} - \frac{(\Sigma_j v_j n_{+jk})^2}{n_{++k}}].$$

To summarize the correlation information from the K strata, Mantel (1963) proposed the statistic

$$M^2 = \frac{\{\Sigma_k[\Sigma_i\Sigma_j u_i v_j n_{ijk} - E(\Sigma_i\Sigma_j u_i v_j n_{ijk})]\}^2}{\Sigma_k \mathrm{Var}(\Sigma_i\Sigma_j u_i v_j n_{ijk})}. \tag{3.6}$$

This is the score statistic for testing conditional independence for model (3.5). It has an asymptotic chi-squared distribution with $df = 1$.

3.2.3 Nominal-by-Ordinal Test

Suppose the row variable X is nominal and the column variable Y is ordinal. A useful loglinear model replaces the ordered row scores in model (3.5) by unordered parameters $\{\mu_i\}$,

$$\log m_{ijk} = \mu + \lambda_i^X + \lambda_j^Y + \lambda_k^Z + \mu_i v_j + \lambda_{ik}^{XZ} + \lambda_{jk}^{YZ}. \tag{3.7}$$

The sufficient statistics for $\{\mu_i\}$ are $\Sigma_j v_j n_{ij+}, i = 1, \cdots, I$. These can be interpreted as the row sums for a response Y within each level of X, using the scores $\{v_j\}$, summed over the strata. Assuming the model holds, we can test conditional independence by testing $\mu_1 = \mu_2 = \cdots = \mu_I$. Let $Y_1, \cdots, Y_{n_{++k}}$ be a random sample within the stratum k, which takes scores $v_1, \cdots, v_J$. Let $\mathbf{l}$ denote the $(I-1) \times 1$ vector having elements

$$l_i = \Sigma_k n_{i+k}(\bar{W}_{ik} - \bar{W}_k), \tag{3.8}$$

where

$$\begin{aligned} \bar{W}_{ik} &= \Sigma_{(h:I_h=i)} Y_h / n_{i+k} \\ &= \Sigma_j n_{ijk} v_j / n_{i+k},\ h = 1, \cdots, n_{++k}, \end{aligned}$$

and

$$\begin{aligned} \bar{W}_k &= \Sigma_{h=1}^{n_{++k}} Y_h / n_{++k} \\ &= \Sigma_i \Sigma_j n_{ijk} v_j / n_{++k}. \end{aligned}$$

Note that $\bar{W}_{ik}$ is the row mean on Y at level i of X and level k of Z, treating Y as a response with scores $\{v_j\}$. Similarly, $\bar{W}_k$ is the kth stratum mean for Y. Let $\mathbf{A}$ denote the null covariance matrix of $\mathbf{l}$, which has elements

$$\begin{aligned} \mathrm{Cov}(l_i, l_{i'}) &= \Sigma_k\left[\frac{n_{i+k}(\delta_{ii'} n_{++k} - n_{i'+k})}{n_{++k}(n_{++k}-1)} \Sigma_{h=1}^{n_{++k}} (W_h - \bar{W}_k)^2\right] \\ &= \Sigma_k\left[\frac{n_{i+k}(\delta_{ii'} n_{++k} - n_{i'+k})}{n_{++k}(n_{++k}-1)} \Sigma_j n_{+jk} (v_j - \bar{W}_k)^2\right]. \end{aligned} \tag{3.9}$$

Then the efficient score statistic for testing conditional independence against the alternative of (3.7) is $\mathbf{l}'\mathbf{A}^{-1}\mathbf{l}$. This statistic is sensitive to location differences among the I conditional distributions of Y that are similar at each level of Z. The asymptotic null distribution is chi-squared with $df = I - 1$.

The three statistics just discussed were suggested by Birch (1965) for testing conditional independence. The three asymptotic tests are available in SAS (PROC FREQ).

3.2.4 Generalized Tests

The previous three statistics are special cases of a general statistic proposed by Landis *et al.* (1978). Let $\mathbf{n}_k$ denote a column vector of the cell counts in stratum k, and let $\mathbf{m}_k$ denote their expected values. Also let P_{i+k} denote the marginal proportion of ith row and let P_{+jk} denote the marginal proportion of jth column. We introduce the following notation to define the generalized test statistic.

$$\boldsymbol{n}'_{ik} = (n_{i1k}, \cdots, n_{iJk})$$

$$\boldsymbol{n}'_k = (\boldsymbol{n}'_{1k}, \cdots, \boldsymbol{n}'_{Ik})$$

$$P_{i+k} = n_{i+k}/n_{++k}$$

$$P_{+jk} = n_{+jk}/n_{++k}$$

$$\boldsymbol{P}'_{*+k} = (P_{1+k}, P_{2+k}, \cdots, P_{I+k}) = (\frac{n_{1+k}}{n_{++k}}, \frac{n_{2+k}}{n_{++k}}, \cdots, \frac{n_{I+k}}{n_{++k}})$$

$$\boldsymbol{P}'_{+*k} = (P_{+1k}, P_{+2k}, \cdots, P_{+Jk}) = (\frac{n_{+1k}}{n_{++k}}, \frac{n_{+2k}}{n_{++k}}, \cdots, \frac{n_{+Jk}}{n_{++k}})$$

Assume that cell counts from different strata are independent. Landis *et al.* (1978) showed that under the hypothesis of conditional independence, the expected value and covariance matrix of the frequencies are, respectively,

$$\boldsymbol{m}_k = E[\boldsymbol{n}_k|H_0] = n_{++k}(\boldsymbol{P}_{*+k} \otimes \boldsymbol{P}_{+*k}) \tag{3.10}$$

and

$$\text{Var}[\boldsymbol{n}_k|H_0] = \frac{n_{++k}^2}{n_{++k}-1}[(\boldsymbol{D}_{\boldsymbol{P}_{*+k}} - \boldsymbol{P}_{*+k}\boldsymbol{P}'_{*+k}) \otimes (\boldsymbol{D}_{\boldsymbol{P}_{+*k}} - \boldsymbol{P}_{+*k}\boldsymbol{P}'_{+*k})], \tag{3.11}$$

where $\otimes$ denotes Kronecker product multiplication and $\boldsymbol{D_a}$ is a matrix with elements of $\boldsymbol{a}$ on the main diagonal.

The generalized statistic for testing conditional independence is defined as

$$Q_M = \boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{G}}^{-1}\boldsymbol{G}, \tag{3.12}$$

where

$$\begin{aligned} \boldsymbol{G} &= \Sigma_k \boldsymbol{B}_k(\boldsymbol{n}_k - \boldsymbol{m}_k) \\ \boldsymbol{V}_{\boldsymbol{G}} &= \Sigma_k \boldsymbol{B}_k[\text{Var}(\boldsymbol{n}_k|H_0)]\boldsymbol{B}'_k, \end{aligned}$$

and where

$$\boldsymbol{B}_k = \boldsymbol{R}_k \otimes \boldsymbol{C}_k$$

is a matrix of fixed constants based on row scores $\boldsymbol{R}_k$ and column scores $\boldsymbol{C}_k$ for the kth stratum. When the null hypothesis is true, the statistic Q_M is approximately distributed as chi-squared with degree of freedom equal to the rank of $\boldsymbol{B}_k$.

Suppose the row variable X is nominal and the column variable Y is ordinal. Then mean score of Y is meaningful. In this case, the mean score is computed for each row of the table, and the alternative hypothesis is that, for at least one stratum, the mean scores of the I rows are unequal. Then the statistic is sensitive to location differences among the I distributions of Y.

For this case we can define the matrix $\boldsymbol{R}_k$ that has dimension $(I-1) \times I$ as

$$\boldsymbol{R}_k = (\boldsymbol{I}_{I-1}, -\boldsymbol{J}_{I-1}), \tag{3.13}$$

where $\boldsymbol{I}_{I-1}$ is an identity matrix of rank $I-1$, and $\boldsymbol{J}_{I-1}$ is an an $(I-1) \times 1$ vector of ones. The matrix has the effect of forming $I-1$ independent contrasts of I mean scores. The matrix $\boldsymbol{C}_k$ has dimension $1 \times J$, and the scores are specified as one for each column. Then Q_M sums over the K strata information about how I row means compare to their null expected values, and it has $df = I - 1$.

When both variables are ordinal, $\boldsymbol{R}_k$ and $\boldsymbol{C}_k$ can be defined as $\boldsymbol{R}_k = (u_1, \cdots, u_I)$, and $\boldsymbol{C}_k = (v_1, \cdots, v_J)$. If the scores $\boldsymbol{R}_k$ and $\boldsymbol{C}_k$ are the same for all strata, Q_M simplifies to M^2.

When both variables are nominal, $\boldsymbol{R}_k = (\boldsymbol{I}_{I-1}, -\boldsymbol{J}_{I-1})$, and $\boldsymbol{C}_k = (\boldsymbol{I}_{J-1}, -\boldsymbol{J}_{J-1})$ can be used. Then Q_M simplifies to $\boldsymbol{d}'\boldsymbol{V}^{-1}\boldsymbol{d}$ with $df = (I-1)(J-1)$.

For exact tests of conditional independence in $I \times J \times K$ tables, we discussed test statistics assuming a lack of three-factor interaction. These are score statistics for loglinear models that treat none, one, or both of the classifications as ordinal. Also they have asymptotic chi-squared distributions.

3.3 Tests of Conditional Independence Permitting Three-factor Interaction

The tests discussed so far assume no three-factor interaction. Suppose, instead, we expect the nature of the association between X and Y to vary considerably across levels of Z. Then one would test against an alternative that permits the association to vary across the strata of Z.

3.3.1 Nominal-by-Nominal Test

Suppose X and Y are nominal. Then one could test conditional independence against the saturated loglinear model, since the only more general model is the saturated model. An efficient score statistic is the Pearson statistic for testing conditional independence against the alternative of the saturated model (Agresti 1992). Letting X_k^2 denote the Pearson statistic for testing independence within the kth level of Z, this statistic is $\Sigma_k X_k^2$. The asymptotic distribution of this statistic is chi-squared with $df = K(I-1)(J-1)$, since at each partial table X_k^2 has asymptotic chi-squared distribution with $df = (I-1)(J-1)$, and we have K independent partial tables. Also, this is the df for testing a loglinear model of conditional independence against the most general alternative.

3.3.2 Ordinal-by-Ordinal Test

The model of homogeneous linear-by-linear association (3.5) allows association between two ordinal variables in each table and this association is homogeneous across levels of Z. When X and Y are ordinal, one sometimes expects a monotone association between X and Y that changes strength across levels of Z. We consider a loglinear model that permits association between X and Y within each level of Z, but heterogeneity among levels of Z, and the degree of heterogeneity is explained by its association parameter. A relevant loglinear model is then the heterogeneous linear-by-linear association model,

$$\log m_{ijk} = \mu + \lambda_i^X + \lambda_j^Y + \lambda_k^Z + \beta_k u_i v_j + \lambda_{ik}^{XZ} + \lambda_{jk}^{YZ}. \tag{3.14}$$

For this model, the null hypothesis of conditional independence is $H_0 : \beta_1 = \cdots = \beta_K = 0$. The loglikelihood is

$$\begin{aligned}
L(\mathbf{m}) &= \sum_i \sum_j \sum_k n_{ijk} \log m_{ijk} - \sum_i \sum_j \sum_k m_{ijk} \\
&= \sum_i \sum_j \sum_k n_{ijk}(\mu + \lambda_i^X + \lambda_j^Y + \lambda_k^Z + \beta_k u_i v_j + \lambda_{ik}^{XZ} + \lambda_{jk}^{YZ}) - \sum_i \sum_j \sum_k m_{ijk} \\
&= n\mu + \sum_i \lambda_i^X n_{i++} + \sum_j \lambda_j^Y n_{+j+} + \sum_k \lambda_k^Z n_{++k} + \sum_k \beta_k \sum_i \sum_j u_i v_j n_{ijk} \\
&+ \sum_i \sum_k \lambda_{ik}^{XZ} n_{i+k} + \sum_j \sum_k \lambda_{jk}^{YZ} n_{+jk} - \sum_i \sum_j \sum_k m_{ijk}. \qquad (3.15)
\end{aligned}$$

For this model the sufficient statistic for β_k is $\Sigma_i \Sigma_j u_i v_j n_{ijk}$. For $k = 1, \cdots, K$, the derivative of the loglikelihood is

$$\frac{\partial L(\mathbf{m})}{\partial \beta_k} = \sum_i \sum_j u_i v_j n_{ijk} - \sum_i \sum_j u_i v_j m_{ijk}.$$

Under the hypothesis of conditional independence, we have $\hat{m}_{ijk} = \frac{n_{i+k} n_{+jk}}{n_{++k}}$. Hence, for $k = 1, \cdots, K$,

$$\begin{aligned}
\frac{\partial L(\mathbf{m})}{\partial \beta_k}\Big|_{m_{ijk}=\hat{m}_{ijk}} &= \sum_i \sum_j u_i v_j (n_{ijk} - \hat{m}_{ijk}) \\
&= \sum_i \sum_j u_i v_j (n_{ijk} - \frac{n_{i+k} n_{+jk}}{n_{++k}}) \\
&= n \sum_i \sum_j u_i v_j (p_{ijk} - \frac{p_{i+k} p_{+jk}}{p_{++k}}).
\end{aligned}$$

Let $\mathbf{s}$ denote the $K \times 1$ vector having elements

$$\begin{aligned}
s_k &= \Sigma_i \Sigma_j u_i v_j (p_{ijk} - \frac{p_{i+k} p_{+jk}}{p_{++k}}) \\
&= \frac{1}{n} \Sigma_i \Sigma_j u_i v_j (n_{ijk} - \frac{n_{i+k} n_{+jk}}{n_{++k}}). \qquad (3.16)
\end{aligned}$$

Then $\mathbf{s}$ can be defined as

$$\mathbf{s} = \begin{bmatrix} \sum_i \sum_j u_i v_j (p_{ij1} - \frac{p_{i+1}p_{+j1}}{p_{++1}}) \\ \sum_i \sum_j u_i v_j (p_{ij2} - \frac{p_{i+2}p_{+j2}}{p_{++2}}) \\ \cdots \\ \sum_i \sum_j u_i v_j (p_{ijk} - \frac{p_{i+k}p_{+jk}}{p_{++k}}) \\ \cdots \\ \sum_i \sum_j u_i v_j (p_{ijK} - \frac{p_{i+k}p_{+jK}}{p_{++K}}) \end{bmatrix}$$

$$= \frac{1}{n} \begin{bmatrix} \sum_i \sum_j u_i v_j (n_{ij1} - \frac{n_{i+1}n_{+j1}}{n_{++1}}) \\ \sum_i \sum_j u_i v_j (n_{ij2} - \frac{n_{i+2}n_{+j2}}{n_{++2}}) \\ \cdots \\ \sum_i \sum_j u_i v_j (n_{ijk} - \frac{n_{i+k}n_{+jk}}{n_{++k}}) \\ \cdots \\ \sum_i \sum_j u_i v_j (n_{ijK} - \frac{n_{i+K}n_{+jK}}{n_{++K}}) \end{bmatrix}.$$

For fixed k, let $\mathbf{G}_k(\boldsymbol{\pi}) = \sum_i \sum_j u_i v_j (\pi_{ijk} - \frac{\pi_{i+k}\pi_{+jk}}{\pi_{++k}})$. Let $\mathbf{g}_k$ represent the $IJ \times 1$ vector having elements

$$\mathbf{g}_k(i,j) = \frac{1}{n_{++k}^2}[(u_i n_{++k} - \Sigma_a u_a n_{a+k})(v_j n_{++k} - \Sigma_b v_b n_{+bk})],$$

and let $\mathbf{g}_k^D$ be the $IJK \times 1$ vector with $\mathbf{g}_k^{D\prime} = (\mathbf{0}'_{(k-1)IJ}, \mathbf{g}'_k, \mathbf{0}'_{(K-k)IJ})$. For example,

$$\mathbf{g}_1^D \equiv \frac{\partial G_1(\boldsymbol{\pi})}{\partial \boldsymbol{\pi}}$$

$$= \frac{1}{\pi_{++1}^2} \begin{bmatrix} (u_1\pi_{++1} - \sum_a u_a \pi_{a+1})(v_1\pi_{++1} - \sum_b v_b \pi_{+b1}) \\ (u_1\pi_{++1} - \sum_a u_a \pi_{a+1})(v_2\pi_{++1} - \sum_b v_b \pi_{+b1}) \\ \cdots \\ (u_i\pi_{++1} - \sum_a u_a \pi_{a+1})(v_j\pi_{++1} - \sum_b v_b \pi_{+b1}) \\ \cdots \\ (u_I\pi_{++1} - \sum_a u_a \pi_{a+1})(v_J\pi_{++1} - \sum_b v_b \pi_{+b1}) \\ \mathbf{0}_{(K-1)IJ} \end{bmatrix}$$

$$
= \frac{1}{n_{++1}^2}\begin{bmatrix} (u_1 n_{++1} - \sum_a u_a n_{a+1})(v_1 n_{++1} - \sum_b v_b n_{+b1}) \\ (u_1 n_{++1} - \sum_a u_a n_{a+1})(v_2 n_{++1} - \sum_b v_b n_{+b1}) \\ \cdots \\ (u_i n_{++1} - \sum_a u_a n_{a+1})(v_j n_{++1} - \sum_b v_b n_{+b1}) \\ \cdots \\ (u_I n_{++1} - \sum_a u_a n_{a+1})(v_J n_{++1} - \sum_b v_b n_{+b1}) \\ \mathbf{0}_{(K-1)IJ} \end{bmatrix}
$$

$$
= \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{0}_{(K-1)IJ} \end{bmatrix},
$$

and

$$
\mathbf{g}_k^D \equiv \frac{\partial G_k(\boldsymbol{\pi})}{\partial \boldsymbol{\pi}}
$$

$$
= \frac{1}{\pi_{++k}^2}\begin{bmatrix} \mathbf{0}_{(k-1)IJ} \\ (u_1 \pi_{++k} - \sum_a u_a \pi_{a+k})(v_1 \pi_{++k} - \sum_b v_b \pi_{+bk}) \\ (u_1 \pi_{++k} - \sum_a u_a \pi_{a+k})(v_2 \pi_{++k} - \sum_b v_b \pi_{+bk}) \\ \cdots \\ (u_i \pi_{++k} - \sum_a u_a \pi_{a+k})(v_j \pi_{++k} - \sum_b v_b \pi_{+bk}) \\ \cdots \\ (u_I \pi_{++k} - \sum_a u_a \pi_{a+k})(v_J \pi_{++k} - \sum_b v_b \pi_{+bk}) \\ \mathbf{0}_{(K-k)IJ} \end{bmatrix}
$$

$$
= \frac{1}{n_{++k}^2}\begin{bmatrix} \mathbf{0}_{(k-1)IJ} \\ (u_1 n_{++k} - \sum_a u_a n_{a+k})(v_1 n_{++k} - \sum_b v_b n_{+bk}) \\ (u_1 n_{++k} - \sum_a u_a n_{a+k})(v_2 n_{++k} - \sum_b v_b n_{+bk}) \\ \cdots \\ (u_i n_{++k} - \sum_a u_a n_{a+k})(v_j n_{++k} - \sum_b v_b n_{+bk}) \\ \cdots \\ (u_I n_{++k} - \sum_a u_a n_{a+k})(v_J n_{++k} - \sum_b v_b n_{+bk}) \\ \mathbf{0}_{(K-k)IJ} \end{bmatrix}
$$

$$
= \begin{bmatrix} \mathbf{0}_{(k-1)IJ} \\ \mathbf{g}_k \\ \mathbf{0}_{(K-k)IJ} \end{bmatrix}.
$$

Also let $\mathbf{D}$ represent the $K \times IJK$ matrix such that row k consists of $\mathbf{g}_k^{D\prime}$, that is

$$\mathbf{D} = \begin{bmatrix} \frac{\partial G_1(\boldsymbol{\pi})'}{\partial \boldsymbol{\pi}} \\ \cdots \\ \frac{\partial G_k(\boldsymbol{\pi})'}{\partial \boldsymbol{\pi}} \\ \cdots \\ \frac{\partial G_K(\boldsymbol{\pi})'}{\partial \boldsymbol{\pi}} \end{bmatrix}.$$

The null asymptotic covariance matrix of $\mathbf{s}$ is $\mathbf{H} = \mathbf{D}\boldsymbol{\Sigma}\mathbf{D}'/n$, where $n = \Sigma\Sigma\Sigma n_{ijk}$ and $\boldsymbol{\Sigma} = Diag(\mathbf{p}) - \mathbf{p}\mathbf{p}'$ with $\mathbf{p} = \{\frac{n_{i+k}n_{+jk}}{nn_{++k}}\}$. The score statistic for testing $H_0 : \beta_1 = \cdots = \beta_K = 0$ is then $\mathbf{s}'\mathbf{H}^{-1}\mathbf{s}$. From Rao (1973, page 418), the asymptotic distribution of $\mathbf{s}$ is K-variate normal. Its mean is zero and dispersion matrix is the information matrix. Hence the asymptotic distribution of $\mathbf{s}'\mathbf{H}^{-1}\mathbf{s}$ is chi-squared with $df = K$. The number of df is the number of components of parameters for testing, or the rank of the asymptotic covariance matrix.

3.3.3 Nominal-by-Ordinal Test

A loglinear model (3.7) implies there are row effects on the association, and these row effects are the same for each level of Z. In general cases when X is nominal and Y is ordinal, we might expect heterogeneity in the row effects on the association. Then a relevant loglinear model to allow heterogeneity across the strata is

$$\log m_{ijk} = \mu + \lambda_i^X + \lambda_j^Y + \lambda_k^Z + \mu_{ik}v_j + \lambda_{ik}^{XZ} + \lambda_{jk}^{YZ}. \tag{3.17}$$

The model is sensitive to alternatives whereby means on Y vary across levels of both X and Z. For identifiability, we use constraints $\sum_i \mu_{ik} = 0$. For this model, the null hypothesis of conditional independence is $H_0 : \mu_{ik} = 0$ for $i = 1, \cdots, I-1$ and

$k = 1, \cdots, K$. The loglikelihood is

$$\begin{aligned}
L(\mathbf{m}) &= \sum_i \sum_j \sum_k n_{ijk} \log m_{ijk} - \sum_i \sum_j \sum_k m_{ijk} \\
&= \sum_i \sum_j \sum_k n_{ijk}(\mu + \lambda_i^X + \lambda_j^Y + \lambda_k^Z + \mu_{ik} v_j + \lambda_{ik}^{XZ} + \lambda_{jk}^{YZ}) - \sum_i \sum_j \sum_k m_{ijk} \\
&= n\mu + \sum_i \lambda_i^X n_{i++} + \sum_j \lambda_j^Y n_{+j+} + \sum_k \lambda_k^Z n_{++k} + \sum_i \sum_j \sum_k \mu_{ik} v_j n_{ijk} \\
&+ \sum_i \sum_k \lambda_{ik}^{XZ} n_{i+k} + \sum_j \sum_k \lambda_{jk}^{YZ} n_{+jk} - \sum_i \sum_j \sum_k m_{ijk}. \qquad (3.18)
\end{aligned}$$

For this model the sufficient statistic for μ_{ik} is $\Sigma_j v_j n_{ijk}$. For fixed i and k, the derivative of the loglikelihood is

$$\frac{\partial L(\mathbf{m})}{\partial \mu_{ik}} = \sum_j v_j n_{ijk} - \sum_j v_j m_{ijk}.$$

Under the hypothesis of conditional independence, we have $\hat{m}_{ijk} = \frac{n_{i+k} n_{+jk}}{n_{++k}}$. Hence, for fixed i and k,

$$\begin{aligned}
\frac{\partial L(\mathbf{m})}{\partial \mu_{ik}}\Big|_{m_{ijk}=\hat{m}_{ijk}} &= \sum_j v_j (n_{ijk} - \hat{m}_{ijk}) \\
&= \sum_j v_j (n_{ijk} - \frac{n_{i+k} n_{+jk}}{n_{++k}}) \\
&= n \sum_j v_j (p_{ijk} - \frac{p_{i+k} p_{+jk}}{p_{++k}}).
\end{aligned}$$

For $i = 1, \cdots, I-1$, $k = 1, \cdots, K$, let $\mathbf{q}$ be the $K(I-1) \times 1$ vector having elements

$$\begin{aligned}
q_{ik} &= \sum_j v_j (p_{ijk} - \frac{p_{i+k} p_{+jk}}{p_{++k}}), \\
&= \frac{1}{n} \sum_j v_j (n_{ijk} - \frac{n_{i+k} n_{+jk}}{n_{++k}}), \\
&= \frac{1}{n} n_{i+k} (\bar{W}_{ik} - \bar{W}_k), \qquad (3.19)
\end{aligned}$$

where $\bar{W}_{ik} = \sum_j n_{ijk}v_j/n_{i+k}$, and $\bar{W}_k = \sum_i \sum_j n_{ijk}v_j/n_{++k}$. Then $\mathbf{q}$ can be defined as

$$\mathbf{q} = \begin{bmatrix} \sum_j v_j(p_{1j1} - \frac{p_{1+1}p_{+j1}}{p_{++1}}) \\ \sum_j v_j(p_{2j1} - \frac{p_{2+1}p_{+j1}}{p_{++1}}) \\ \cdots \\ \sum_j v_j(p_{(I-1)j1} - \frac{p_{(I-1)+1}p_{+j1}}{p_{++1}}) \\ \cdots\cdots\cdots\cdots \\ \cdots\cdots\cdots\cdots \\ \sum_j v_j(p_{1jk} - \frac{p_{1+k}p_{+jk}}{p_{++k}}) \\ \sum_j v_j(p_{2jk} - \frac{p_{2+k}p_{+jk}}{p_{++k}}) \\ \cdots \\ \sum_j v_j(p_{(I-1)jk} - \frac{p_{(I-1)+k}p_{+jk}}{p_{++k}}) \\ \cdots\cdots\cdots\cdots \\ \cdots\cdots\cdots\cdots \\ \sum_j v_j(p_{1jK} - \frac{p_{1+K}p_{+jK}}{p_{++K}}) \\ \sum_j v_j(p_{2jK} - \frac{p_{2+K}p_{+jK}}{p_{++K}}) \\ \cdots \\ \sum_j v_j(p_{(I-1)jK} - \frac{p_{(I-1)+K}p_{+jK}}{p_{++K}}) \end{bmatrix}.$$

Or it can be written as

$$\mathbf{q} = \frac{1}{n} \begin{bmatrix} \sum_j v_j(n_{1j1} - \frac{n_{1+1}n_{+j1}}{n_{++1}}) \\ \sum_j v_j(n_{2j1} - \frac{n_{2+1}n_{+j1}}{n_{++1}}) \\ \cdots \\ \sum_j v_j(n_{(I-1)j1} - \frac{n_{(I-1)+1}n_{+j1}}{n_{++1}}) \\ \cdots\cdots\cdots\cdots \\ \cdots\cdots\cdots\cdots \\ \sum_j v_j(n_{1jk} - \frac{n_{1+k}n_{+jk}}{n_{++k}}) \\ \sum_j v_j(n_{2jk} - \frac{n_{2+k}n_{+jk}}{n_{++k}}) \\ \cdots \\ \sum_j v_j(n_{(I-1)jk} - \frac{n_{(I-1)+k}n_{+jk}}{n_{++k}}) \\ \cdots\cdots\cdots\cdots \\ \cdots\cdots\cdots\cdots \\ \sum_j v_j(n_{1jK} - \frac{n_{1+K}n_{+jK}}{n_{++K}}) \\ \sum_j v_j(n_{2jK} - \frac{n_{2+K}n_{+jK}}{n_{++K}}) \\ \cdots \\ \sum_j v_j(n_{(I-1)jK} - \frac{n_{(I-1)+K}n_{+jK}}{n_{++K}}) \end{bmatrix}.$$

For fixed i, k, let $G_{ik}(\boldsymbol{\pi}) = \sum_j v_j(\pi_{ijk} - \frac{\pi_{i+k}\pi_{+jk}}{\pi_{++k}})$. Let $\mathbf{r}_{ik}$ represent the $IJ \times 1$ vector having elements

$$\mathbf{r}_{ik}(i', j) = \frac{1}{n_{++k}^2}[(v_j n_{++k} - \Sigma_b v_b n_{+bk})(n_{++k}\delta_{ii'} - n_{i+k})],\ i' = 1, \cdots, I,$$

and let $\mathbf{r}_{ik}^E$ be the $IJK \times 1$ vector with $\mathbf{r}_{ik}^{E\prime} = (\mathbf{0}'_{(k-1)IJ}, \mathbf{r}'_{ik}, \mathbf{0}'_{(K-k)IJ})$. That is,

$$\begin{aligned}
\mathbf{r}_{ik}^E &\equiv \frac{\partial G_{ik}(\boldsymbol{\pi})}{\partial \boldsymbol{\pi}} \\
&= \frac{1}{\pi_{++k}^2}\left[\begin{array}{c}
\mathbf{0}_{(k-1)IJ} \\
(v_1\pi_{++k} - \sum_b v_b\pi_{+bk})(-\pi_{i+k}) \\
(v_2\pi_{++k} - \sum_b v_b\pi_{+bk})(-\pi_{i+k}) \\
\cdots \\
(v_J\pi_{++k} - \sum_b v_b\pi_{+bk})(-\pi_{i+k}) \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
(v_1\pi_{++k} - \sum_b v_b\pi_{+bk})(\pi_{++k} - \pi_{i+k}) \\
(v_2\pi_{++k} - \sum_b v_b\pi_{+bk})(\pi_{++k} - \pi_{i+k}) \\
\cdots \\
(v_J\pi_{++k} - \sum_b v_b\pi_{+bk})(\pi_{++k} - \pi_{i+k}) \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
(v_1\pi_{++k} - \sum_b v_b\pi_{+bk})(-\pi_{i+k}) \\
(v_2\pi_{++k} - \sum_b v_b\pi_{+bk})(-\pi_{i+k}) \\
\cdots \\
(v_J\pi_{++k} - \sum_b v_b\pi_{+bk})(-\pi_{i+k}) \\
\mathbf{0}_{(K-k)IJ}
\end{array}\right].
\end{aligned}$$

Or

$$
\mathbf{r}_{ik}^{E} = \frac{1}{n_{++k}^2}\begin{bmatrix}
\mathbf{0}_{(k-1)IJ} \\
(v_1 n_{++k} - \sum_b v_b n_{+bk})(-n_{i+k}) \\
(v_2 n_{++k} - \sum_b v_b n_{+bk})(-n_{i+k}) \\
\cdots \\
(v_J n_{++k} - \sum_b v_b n_{+bk})(-n_{i+k}) \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
(v_1 n_{++k} - \sum_b v_b n_{+bk})(n_{++k} - n_{i+k}) \\
(v_2 n_{++k} - \sum_b v_b n_{+bk})(n_{++k} - n_{i+k}) \\
\cdots \\
(v_J n_{++k} - \sum_b v_b n_{+bk})(n_{++k} - n_{i+k}) \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
(v_1 n_{++k} - \sum_b v_b n_{+bk})(-n_{i+k}) \\
(v_2 n_{++k} - \sum_b v_b n_{+bk})(-n_{i+k}) \\
\cdots \\
(v_J n_{++k} - \sum_b v_b n_{+bk})(-n_{i+k}) \\
\mathbf{0}_{(K-k)IJ}
\end{bmatrix}
\equiv \begin{bmatrix} \mathbf{0}_{(K-1)IJ} \\ \mathbf{r}_{ik} \\ \mathbf{0}_{(K-k)IJ} \end{bmatrix}.
$$

Also let $\mathbf{E}$ represent the $K(I-1) \times IJK$ matrix such that the row corresponding to i, k consists of $\mathbf{r}_{ik}^{E\prime}$, that is,

$$
\mathbf{E} = \begin{bmatrix}
\frac{\partial G_{11}(\boldsymbol{\pi})'}{\partial \boldsymbol{\pi}} \\
\cdots \\
\frac{\partial G_{(I-1)1}(\boldsymbol{\pi})'}{\partial \boldsymbol{\pi}} \\
\cdots\cdots\cdots \\
\cdots\cdots\cdots \\
\frac{\partial G_{1K}(\boldsymbol{\pi})'}{\partial \boldsymbol{\pi}} \\
\cdots \\
\frac{\partial G_{(I-1)K}(\boldsymbol{\pi})'}{\partial \boldsymbol{\pi}}
\end{bmatrix}.
$$

The null asymptotic covariance matrix of $\mathbf{q}$ is $\mathbf{R} = \mathbf{E\Sigma E}'/n$. The score statistic for

testing $H_0 : \mu_{ik} = 0$ for $i = 1, \cdots, I-1$ and $k = 1, \cdots, K$ is $\mathbf{q}'\mathbf{R}^{-1}\mathbf{q}$. Its asymptotic distribution is chi-squared with $df = K(I-1)$. The number of df is the rank of the asymptotic covariance matrix or the number of components of parameters for testing.

For exact tests, one identifies any of these six statistics with T in the calculation of the exact P-value. We discuss next how to construct modified exact P-values for the six tests.

3.4 The Construction of the Modified Exact P-value

So far, we have discussed six test statistics for testing conditional independence of X and Y, given Z, in three-way contingency tables. The ordinary exact P-value can be constructed by utilizing these statistics. In Chapter 2, we proposed a modified exact P-value, to reduce the degree of conservativeness. It is based on both the usual test statistic and, at the observed value of T, a secondary statistic T' that generates a secondary partitioning. The statistic T' is a statistic directed toward a broader alternative. Then, T' can catch some information about the validity of the null hypothesis when the assumed alternative for T is not exactly satisfied. The modified exact P-value is defined in Chapter 2 as

$$P^* = P_{H_0}(T > t_o) + P_{H_0}(T = t_o,\, T' \geq t'_o),$$

when large values of T and T' contradict the null. We have shown in Chapter 2, using $2 \times 2 \times K$ tables, that the modified P-value has less discrete sampling distributions, and modified tests reduce the degree of conservativeness. We can apply this modified approach to $I \times J \times K$ tables to reduce the conservativeness and to get sharper results.

For testing conditional independence assuming no three-factor interaction, we denote T_1 to be the test statistic when both X and Y are nominal, denote T_2 to be

the test statistic when X is nominal and Y is ordinal, denote T_2' to be the test statistic when X is ordinal and Y is nominal, and denote T_3 to be the test statistic when both X and Y are ordinal. Also, let T_4, T_5, T_5' and T_6 be the corresponding test statistics when we permit three-factor interaction. Note that these are score statistics.

In this section, we discuss possible alternative ways of forming modified P-values for testing conditional independence for $I \times J \times K$ tables. Ordinary exact P-values for these six tests correspond to six loglinear models for primary alternative hypotheses. The general rule to construct the modified exact P-value is as follows. We use a score statistic for T', in order to have consistency. If there is only one potential statistic for T', we use that one. But, if there is more than one potential statistic, we apply a basic principle to choose a T' among them. Now, we establish basic principles. We can consider four types of principles. The first principle is to choose a T' from the next most general alternative, while keeping the same assumption as T about three-factor interaction. The second principle is to choose a T' from the most general alternative, while keeping the same assumption as T about three-factor interaction. The third principle is to choose a T' from the most general alternative among all cases. The fourth principle is to choose a T' while keeping the nature of the classification variables. Next, we discuss all possible statistics for T' for six cases. Note that all possible potential statistics for T' are $T_1, T_2, T_2', T_3, T_4, T_5, T_5'$, and T_6. We first consider the tests assuming no three-factor interaction.

When both X and Y are nominal, the primary test statistic T is T_1. The secondary statistic T' can be T_4, since T_4 corresponds to a more general alternative hypothesis. Second, when X is nominal and Y is ordinal, T is T_2 and T' can be T_1, T_4, or T_5. Third, when both X and Y are ordinal, T is T_3 and T' can be $T_1, T_2, T_2', T_4, T_5, T_5'$, or T_6. Since T_3 is constructed from the narrowest alternative, the other statistics can be potential statistics for T'.

Next, we assume three-factor interaction. First, when both X and Y are nominal, T is T_4, but there is no general score statistic for T', since T is constructed from the most general alternative. We could, however, use the table probability for T' for the secondary partitioning. Second, when X is nominal and Y is ordinal, T is T_5 and T' can be T_4. Finally, when both X and Y are ordinal, T is T_6, and T' can be T_4, T_5 or T_5'. Table 3.1 summarizes all possible statistics for T' for six tests.

We see two cases have only one potential statistic for T'. For the nominal-by-nominal case assuming no three-factor interaction, T' is T_4. Note that permitting three-factor interaction, nominal-by-nominal case, there is no score statistic, but we could use the table probability. Also, for the nominal-by-ordinal case, T' is T_4. For these three cases, there is only one choice for T'. For other three cases, we apply a basic principle in order to choose a T' among potential statistics.

For the first principle, we choose a T' from the next most general alternative, while keeping the same assumption as T about three-factor interaction. Assuming no-three factor interaction, (T, T') is (T_2, T_1) for the nominal-by-ordinal case, since the nominal-by-nominal case is more general, and it also corresponds to the next most general alternative assuming no three-factor interaction in this case. For the ordinal-by-ordinal case, the next most general alternative corresponds to the nominal-by-ordinal case or the ordinal-by nominal case. Hence (T, T') is (T_3, T_2) or (T_3, T_2'). Accordingly, for the ordinal-by-ordinal case permitting three-factor interaction, (T, T') is (T_6, T_5) or (T_6, T_5').

The second principle is to choose a T' from the most general alternative among three cases, while keeping the same assumption as T about three-factor interaction. Then, assuming no-three factor intercation, the corresponding statistics for (T, T') is (T_2, T_1) for the nominal-by-ordinal case and (T_3, T_1) for the ordinal-by-ordinal case, since the nominal-by-nominal case is the most general among three cases. Also, for the ordinal-by-ordinal case permitting three-factor intercation, (T, T') is (T_6, T_4).

For the third principle of the most general alternative among all cases, the corresponding statistics for (T, T') is $(T_2, T_4), (T_3, T_4)$, and (T_6, T_4), since T_4 corresponds to the most general alternative among all cases. For the fourth principle of keeping the nature of the classification variables, the corresponding statistics for (T, T') is $(T_2, T_5), (T_3, T_6)$. For the ordinal-by-ordinal case permitting three-factor interaction, T' does not have a potential statistic in this principle.

Among four principles, we prefer the first principle, since modified P-values can be defined for most cases using this principle, and it can utilize the ordinality of classification variables. For the second and third principles, T' does not consider possible ordinality. Table 3.2 summarizes test statistics for the construction of ordinary and modified exact P-values for testing conditional independence in $I \times J \times K$ contingency tables using the first principle. For $I \times J \times K$ contingency tables, the discreteness will not be severe when the sample size is large. But, when the sample size is small, the modified P-value can reduce the conservativeness. We discuss implementation of the exact tests in the next section.

Table 3.1. All possible statistics for T' for six tests.

	T	T'							
		T_1	T_2	T_2'	T_3	T_4	T_5	T_5'	T_6
Assuming no three-factor interaction									
Nominal-by-Nominal	T_1	·	·	·	·	T_4	·	·	·
Nominal-by-Ordinal	T_2	T_1	·	·	·	T_4	T_5	·	·
Ordinal-by-Ordinal	T_3	T_1	T_2	T_2'	·	T_4	T_5	T_5'	T_6
Permitting three-factor interaction									
Nominal-by-Nominal	T_4	·	·	·	·	·	·	·	·
Nominal-by-Ordinal	T_5	·	·	·	·	T_4	·	·	·
Ordinal-by-Ordinal	T_6	·	·	·	·	T_4	T_5	T_5'	·

Table 3.2. Test statistics for the construction of the ordinary and modified exact P-values P^* for testing conditional independence in $I \times J \times K$ contingency tables.

	Ordinary P-value	Modified P-value P^*
	T	(T, T')
Assuming no three-factor interaction		
Nominal-by-Nominal	T_1	(T_1, T_4)
Nominal-by-Ordinal	T_2	(T_2, T_1)
Ordinal-by-Ordinal	T_3	(T_3, T_2)
Permitting three-factor interaction		
Nominal-by-Nominal	T_4	$(T_4, P(\mathbf{Z}))$
Nominal-by-Ordinal	T_5	(T_5, T_4)
Ordinal-by-Ordinal	T_6	(T_6, T_5)

3.5 Approximation of Exact P-values

For three-way contingency tables, algorithms for testing conditional independence are available in widely-available software only for the $2 \times J \times K$ case with ordered columns (StatXact 1991). Even for table sizes where software exists, the reference set of tables for the conditional distribution is sometimes too large for an exact P-value computation. For instance, sometimes the sample size is moderately large but there are many cells and the table is sparse, so exact methods are infeasible but the use of standard asymptotic theory is questionable.

In some cases, one can obtain a very accurate approximation to the distribution of the test statistic using a saddlepoint approximation. This higher-order asymptotic approximation is more accurate than the normal approximation or the one- or two-term Edgeworth expansion. It is applicable to conditional densities and tail probabilities of sufficient statistics in exponential families. For example, to approximate

conditional tail probabilities, one can use an approximation due to Skovgaard (1987). Davison (1988) applied the approximation to model (3.5) for $2 \times 2 \times K$ tables, and Pierce and Peters (1992) applied it to model (3.5) for $K = 1$.

To illustrate the saddlepoint approximation, we show how to apply it to the homogeneous linear-by-linear association model (3.5) for arbitrary K. Let $\hat{\beta}$ denote the ML estimate of β in that model. Let $G^2(I)$ and $G^2(L \times L)$ denote the likelihood-ratio statistics for testing the goodness of fit of the conditional independence and homogeneous linear-by-linear association models. The conditional P-value for testing $H_0 : \beta = 0$ against $H_1 : \beta > 0$ has saddlepoint approximation

$$Pr(T \geq t_o | \{n_{i+k}\}, \{n_{+jk}\}) \simeq 1 - \Phi(z) + \phi(z)(\frac{1}{w} - \frac{1}{z}), \tag{3.20}$$

where

$$z = \mathrm{sgn}(\hat{\beta})\sqrt{G^2(I) - G^2(L \times L)} \qquad \text{and} \qquad w = 2\sinh(\frac{\hat{\beta}}{2})\sqrt{\frac{|I_{L \times L}|}{|I_I|}}.$$

The matrices I_I and $I_{L \times L}$ are the observed information matrices for the conditional independence model and homogeneous linear-by-linear association model, and Φ and ϕ denote the standard normal *cdf* and *pdf*.

Since software is not yet available in the generality needed for the exact conditional methods we have described for $I \times J \times K$ tables, we next present an alternative method that can approximate the exact conditional result as well as needed. This is the simple approach of performing a Monte Carlo simulation on the conditional set. The Monte Carlo method is an alternative to computing either the exact or asymptotic P-values. It is useful for those situations where the data set is too large for an exact P-value computation or too sparse to rely on the asymptotic theory.

Agresti *et al.* (1979) utilized this method effectively for a variety of tests for two-way tables. Even for large tables or large sample sizes, one can quickly approximate

as closely as needed the ordinary and modified exact P-values for the six statistics presented in Section 2 and Section 3. The method consists of sampling contingency tables from the conditional reference set in proportion to their probabilities, and computing an unbiased point estimate and a narrow confidence interval for an exact P-value. We constructed an algorithm to perform precise approximations for the exact inferences using a table-generation procedure suggested by Patefield (1981). For practical applications, we prefer this approximation to the saddlepoint because it is available more generally (*e.g.,* for multi-degree-of-freedom statistics for testing vectors of parameters) because its accuracy is known to the user, and because that accuracy can be set as finely as one requires.

We proposed ordinary and modified exact P-values for six tests, and T and T' are defined in Table 3.2. To illustrate, suppose we want to estimate a modified exact one-sided P-value when X and Y are ordinal assuming no three-factor interaction. Then, we test against a narrower alternative of the homogeneous linear-by-linear association model (3.5). The secondary statistic T' is a test statistic directed toward a broader alternative hypothesis. For T', one possibility is the score statistic for the case of nominal-ordinal association assuming no three-factor interaction. Let t'_o be the observed value of T'. Therefore, in this case we have $T = \sum\sum\sum u_i v_j n_{ijk}$, and T' is a score statistic discussed in Section 3.2.3. This is a one-sided test. Accordingly, modified exact P-values for other tests can be constructed by using T and T' in Table 3.2. They are two-sided tests.

To implement the exact tests, we sample M contingency tables, with replacement, from the reference set Γ of tables with the same margins, where M is chosen to give the desired degree of accuracy with some fixed probability. Define the upper critical region of the reference set by

$$\Gamma^* = \{\mathbf{Z} \in \Gamma : T > t_o \text{ or } (T = t_o \text{ and } T' \geq t'_o)\ \}.$$

The other possibility for T' is to use the null table probability. Under the null hypothesis of conditional independence, the probability of observing any specific $\mathbf{Z} \in \Gamma$ is

$$Pr(\mathbf{Z} = \mathbf{z}) = \Pi_{k=1}^{K} \frac{\Pi_i n_{i+k}! \Pi_j n_{+jk}!}{n_{++k}! \Pi_i \Pi_j n_{ijk}!}. \tag{3.21}$$

Then we define the critical region of the reference set by

$$\Gamma_p^* = \{\mathbf{Z} \in \Gamma : T > t_o \text{ or } (T = t_o \text{ and } P(\boldsymbol{Z}) \leq P(\boldsymbol{N}))\ \}.$$

For the ith table sampled, let $y_i = 1$ if $z_i \in \Gamma^*$, and let $y_i = 0$, otherwise. The point estimate of the modified P-value is

$$\hat{p^*} = \frac{1}{M} \Sigma_i y_i,$$

the proportion of sampled tables in Γ^*. Likewise, the estimate of the modified P-value using the null table probability for T' can be defined using Γ_p^*, and we denote by $\hat{p_p^*}$. For the estimate of ordinary exact P-value, the upper critical region of the reference set, Γ', is

$$\Gamma' = \{\mathbf{Z} \in \Gamma : T \geq t_o\},$$

that is, the proportion of sampled tables that have a test statistic at least as large as the observed one.

3.6 Examples

3.6.1 Example 1

We illustrate the exact tests using Table 3.3. This is a cross classification of job satisfaction by income, controlling for gender, for black Americans sampled in the

General Social Survey of 1991. In order to utilize ordinality in studying the partial association between income and satisfaction, we test conditional independence against the model (3.5) of homogeneous linear-by-linear association. Using equally-spaced row and column scores, the likelihood-ratio chi-squared statistic for testing the fit of that model equals 12.33, with $df = 17$. The estimated association parameter is $\hat{\beta} = 0.388$ with $s.e. = 0.155$. The likelihood-ratio chi-squared statistic for testing conditional independence, assuming the model, is 19.37-12.33=7.04 with $df = 1$. There seems to be very strong evidence of a positive association between income and satisfaction. However, the data are sparse enough to make large-sample approximations questionable; yet the sample size is sufficiently large so that exact analyses are infeasible. We used Monte Carlo sampling with $M = 50,000$, which guarantees that P-value estimators fall within 0.004 of the true P-value with probability at least 0.95.

For the exact tests assuming no three-factor interaction, the estimated exact P-values for the ordinary exact P-values (with 95% precision indicated in parentheses) are 0.332 ($\pm$ 0.004) for the nominal-by-nominal test, 0.024 ($\pm$ 0.001) for the nominal-by-ordinal test, and 0.006 ($\pm$ 0.001) for the ordinal-by-ordinal test. Using T' defined in Table 3.2, the corresponding estimated exact P-values for modified exact P-values P^* are 0.332, 0.024, and 0.004. Also using the null table probability for T', the corresponding estimated modified P-values P_p^* are 0.332, 0.024, and 0.005. The distribution of T takes 121 separate points for the ordinal-by-ordinal test, and since the degree of discreteness is not severe, the two types of P-values are essentially the same. The asymptotic P-values are 0.335, 0.026, and 0.005, respectively. In this case, first-order asymptotic approximations work quite well.

For other exact tests permitting three-factor interaction, the estimated exact P-values for the ordinary exact P-values are 0.281 for the nominal-by-nominal test, 0.089 for the nominal-by-ordinal test, and 0.020 for the ordinal-by-ordinal test. The corresponding estimated P-values for modified exact P-value, P^* or P_p^*, are 0.281,

0.089, and 0.020. Also, the corresponding asymptotic P-values are 0.277, 0.089, and 0.020. Table 3.4 summarizes results for all six tests we have discussed. Note that we would not obtain strong evidence of association if we ignored the ordinality of the variables. For large n, since the discreteness is not severe, the modified approach is not needed. Generally, the modified P-value is less discrete than the ordinary P-value and leads to less conservative tests. For small n, we can see the advantage of using the modified approach.

Table 3.3. Cross-classification of job satisfaction with income, controlling for gender, for black Americans.

Gender	Income	Satisfaction			
		VD	LS	MS	VS
Male	< 5000	1	1	2	1
	< 15000	0	3	5	1
	< 25000	0	0	7	3
	≥ 25000	0	1	9	6
Female	< 5000	1	3	11	2
	< 15000	2	3	17	3
	< 25000	0	1	8	5
	≥ 25000	0	2	4	2

Source: General Social Surveys (1991)
VD : Very Dissatisfied, LS : A little Satisfied
MS : Moderately Satisfied, VS : Very Satisfied

3.6.2 Example 2

We next illustrate the exact tests of independence using Table 3.5, which is a 3×2 table from the example in Table 1 of Patefield (1982). This is the results of a double-blind study concerning the use of Oxprenolol in the treatment of examination stress. Among 32 students, 15 were treated with Oxprenolol and 17 were given Diazepam

Table 3.4. Estimated exact P-values for testing conditional independence in Table 3.3.

	Ordinary P-value	Modified P-value P^*	Modified P-value P_p^*	Asymptotic P-value
Assuming no three-factor interaction				
Nominal-by-Nominal	0.332	0.332	0.332	0.335
Nominal-by-Ordinal	0.024	0.024	0.024	0.026
Ordinal-by-Ordinal	0.006	0.004	0.005	0.005
Permitting three-factor interaction				
Nominal-by-Nominal	0.281	0.281	0.281	0.277
Nominal-by-Ordinal	0.089	0.089	0.089	0.089
Ordinal-by-Ordinal	0.020	0.020	0.020	0.021

(control). The examination results were compared with their tutor's prediction. The column classification is ordinal, and the row classification can be assumed as ordinal since it has two levels.

When X and Y are ordinal, a relevant model that reflects the ordinality in a two-way table is the model of linear-by-linear association,

$$\log m_{ij} = \mu + \lambda_i^X + \lambda_j^Y + \beta u_i v_j. \tag{3.22}$$

The independence model is the special case of $\beta = 0$. We test independence against the model of linear-by-linear association in order to utilize ordinality. For unit-spaced scores, the likelihood-ratio chi-squared statistic for testing the fit of that model equals 2.64, with $df = 1$. The estimated association parameter is $\hat{\beta} = 1.706$ with $s.e. =$ 0.773. The likelihood-ratio chi-squared statistic for testing independence, assuming the model, is 9.38-2.64=6.74 with $df = 1$ (P=0.009). There seems to be very strong evidence that the examination grades compared with their tutor's prediction tend to be higher in the treatment group. Large-sample approximations are questionable

since the sample size is small. We use Monte Carlo sampling with $M = 50,000$ and compare the estimated exact P-value with the the exact P-value.

For the exact tests of independence, the estimated exact P-values for the ordinary exact P-values (with 95% precision indicated in parentheses) are 0.026 ($\pm$ 0.001) for the nominal-by-nominal test, 0.024 ($\pm$ 0.001) for the nominal-by-ordinal test, and 0.013 ($\pm$ 0.001) for the ordinal-by-ordinal test. The corresponding estimated exact P-values for modified exact P-values P^* are 0.026, 0.017, and 0.013. The asymptotic P-values are 0.028, 0.015, and 0.007, respectively. The ordinary exact P-value for the ordinal-by-ordinal test is 0.013. For a $I \times J$ table with ordinal variables, StatXact gives ordinary exact P-values, based on methodology in Agresti *et al.* 1990. Table 3.6 summarizes results for the tests we have discussed. Note that utilizing the ordinality provides very strong evidence of association. Also, the modified P-value can give sharper inference for small n.

Table 3.5. Examination results compared with tutor's predictions.

Group	Results		
	Better	Same	Worse
Treated	5	8	2
Control	0	11	6

Source: Patefield (1982)
Better : Better than predicted
Same : Same as predicted
Worse : Worse than predicted

Table 3.6. Estimated exact P-values for testing independence in Table 3.5.

	Ordinary P-value	Modified P-value P^*	Asymptotic P-value
Nominal-by-Nominal	0.026	0.026	0.028
Nominal-by-Ordinal	0.024	0.017	0.015
Ordinal-by-Ordinal	0.013	0.013	0.007

3.7 FORTRAN Program for Simulation

Patefield (1981) provided a subroutine for generating two-way random tables with fixed row and column totals. We can apply his algorithm stratum by stratum in order to construct three-way random contingency tables. We utilize the six exact tests for testing conditional independence in $I \times J \times K$ contingency tables that were discussed in Section 2 and Section 3. These test statistics are score statistics for loglinear models, and they do not require fitting the model. The computations, which involve simulating exact conditional distributions, are considerably simpler when one can use test statistics that do not require fitting the model for each table generated for the simulations.

Boyett (1979) also constructed a subroutine that generates two-way random tables from the exact distribution with given row and column totals. Patefield's (1981) subroutine is faster for larger values of n, and it can calculate the probability of each generated random table.

By the Monte Carlo sampling of tables in the reference set, we can approximate exact inference with simulated exact and modified exact P-values for testing conditional independence. By resampling these random contingency tables, the P-value is updated. The FORTRAN program runs interactively. For computational accuracy, double precision is used. This program is designed for IBM-compatible PCs or UNIX

workstations, and the general structure of the program and part of FORTRAN source code are listed in Appendix B.

3.7.1 Restrictions

Two-way random tables must have at least two rows and two columns, and row and column totals should be positive. The maximum number of rows and columns is 50, and maximum number of strata is 20. The number $(NROW - 1) \times (NCOL - 1)$ should be less than 250. This is the maximum array for the variance-covariance matrix in the nominal-by-nominal test.

Recursive calculation of log-factorial through $\log(n+1)! = \log(n)! + \log(n+1)$ has the disadvantage of accumulating a large rounding error (Verbeek and Kroonerberg 1985). For accuracy, double precision is used for the log-factorial, and the log-factorial can be computed up to 25000.

CHAPTER 4
IMPROVED EXACT TESTS FOR ORDINAL VARIABLES IN $I \times J \times K$ TABLES

4.1 Introduction

Consider contingency tables under the full multinomial model where row and column classifications are ordinal. In two-way contingency tables when both classifications are ordinal, the null hypothesis of independence can be tested against the alternative that utilizes local log odds ratios. Many tests for measuring ordinal association have been proposed. We can utilize tests based on $C - D$, the number of concordant pairs minus the number of discordant pairs, or based on the gamma statistic. Both are discussed in Agresti (1990). Also, log-linear models with ordered categories are discussed. Agresti, Mehta, and Patel (1990) provide an algorithm that permits exact tests for the linear-by-linear association model for two-way contingency tables with ordered categories.

If an exact test is desired with size being equal to some preassigned value, then randomization would be required on some tables of observed frequencies. This is typical of any discrete problem. We want the resulting test to be admissible even though randomization occurred. Cohen and Sackrowitz (1991) proved a theorem that gives the class of exact, unbiased, and admissible tests. Also, Cohen and Sackrowitz (1992) suggested a procedure for an exact test of size α, and a modified P-value. Such tests are performed conditionally, given the values of the sufficient statistics for the nuisance parameters under the null hypothesis. Hence, the critical value depends on the values of the sufficient statistics.

They constructed the exact test of size α by ordering the tables according to their probabilities on sample points where the test would randomize. They made the number of tables on which randomization would occur considerably smaller than in the usual test. We could use another test statistic directed toward a broader alternative hypothesis at the randomization points, utilizing the modified approach discussed in Chapter 2.

Cohen and Sackrowitz (C-S) focused on two-way tables, and showed unbiasedness of tests in two-way tables. Eaton (1970) showed the essentially complete class in an exponential family. Eaton's theorem shows that the essentially complete class consists of tests whose acceptance regions are convex with possible randomization on the boundary of acceptance region. Furthermore, Ledwina (1978a, 1984) gave the class of admissible rules in an exponential family. Admissibility of tests for the C-S theorem is obtained using the arguments in Ledwina.

We focus on analyzing three-way tables. The problem we will consider is testing conditional independence, assuming that the model of no three-factor interaction holds. We first introduce theorems and lemmas from C-S (1991), and then generalize these to three-way contingency tables. In Section 2 we state the theorem of C-S (1991) as well as related lemmas, that give the class of unbiased admissible tests. In Section 3 we show unbiasedness of tests when one wishes to test a null hypothesis of conditional independence against the alternative of no three-factor interaction model in three-way contingency tables. Sections 4 and 5 present the complete class of tests and admissible tests in an exponential family. Using these arguments, the tests of the C-S theorem lie in a complete and admissible class when we consider three-way tables under the multinomial model.

Section 6 generalizes to the three-way case some results of Cohen and Sackrowitz (1991, 1992) regarding admissibility of tests for two-way tables. For an ordinal alternative, we discuss construction of tests of conditional independence that are exact,

unbiased, and admissible. As a special case, we note that the ordinary randomized test of conditional independence for $2 \times 2 \times K$ tables is usually inadmissible.

Section 7 illustrates the exact, unbiased and admissible tests with examples. We test conditional independence in $2 \times 2 \times 5$ tables. Section 8 gives some comments.

4.2 Basic Results in Two-way Contingency Table

Consider testing independence against the alternative that all local log odds ratios are nonnegative with at least one local log odds ratio positive for a two-way table. We will state the class of tests that are simultaneously exact, unbiased, and admissible in this section. We need definitions and lemmas for the proof of unbiasedness of tests, obtained by Cohen and Sackrowitz (1991). We will extend their theorem and lemmas to three-way tables in the next section.

Consider an $I \times J$ contingency table under the full multinomial model where each classification is ordinal . Let $\boldsymbol{N} = \{n_{ij}\}$ be the $I \times J$ two-way contingency table of cell frequencies, and let $\boldsymbol{\pi} = \{\pi_{ij}\}$ be the $I \times J$ matrix of corresponding cell probabilities, where $n = \Sigma\Sigma n_{ij}$, and $\Sigma\Sigma\pi_{ij} = 1$. Let n_{i+} be the ith row total of cell frequencies, $i = 1, \cdots, I-1$, and n_{+j} the jth column total of cell frequencies, $j = 1, \cdots, J-1$, $\boldsymbol{m} = (\{n_{i+}\}, \{n_{+j}\})$. We define the local log odds ratios as $\psi_{ij} = \log \frac{\pi_{ij}\pi_{i+1,j+1}}{\pi_{i,j+1}\pi_{i+1,j}}$, $i = 1, \cdots, I-1$, $j = 1, \cdots, J-1$. Our testing problem can be expressed as testing the null hypothesis $H_0 : \pi_{ij} = \pi_{i+}\pi_{+j}$ for $i = 1, \cdots, I-1$, $j = 1, \cdots, J-1$. From Ledwina (1984), under the full multinomial model, the distribution of an observed random vector, $\boldsymbol{N}$, of the cell frequencies $(n_{11}, \cdots, n_{IJ})$ can be written in the form

$$f(\boldsymbol{N}) = d^n n! \Pi_{i,j=1}^{IJ} (n_{ij}!)^{-1} \exp(\Sigma_{i,j=1}^{I-1,J-1} n_{ij}a_{ij} + \Sigma_{i=1}^{I-1} n_{i+}b_i + \Sigma_{j=1}^{J-1} n_{+j}d_j), \quad (4.1)$$

where

$$a_{ij} = \log\frac{\pi_{ij}\pi_{IJ}}{\pi_{iJ}\pi_{Ij}}$$

$$b_i = \log\frac{\pi_{iJ}}{\pi_{IJ}}$$

$$d_j = \log\frac{\pi_{Ij}}{\pi_{IJ}}$$

and

$$d = (1 + \Sigma_i^{I-1} e^{bi} + \Sigma_j^{J-1} e^{dj} + \Sigma e^{a_{ij}+b_i+d_j})^{-1}.$$

Note that (4.1) is the density of multivariate exponential family, and $a_{kl} = \Sigma_{i=k}^{I-1}\Sigma_{j=l}^{J-1}\psi_{ij}$. Then our hypotheses become $H_0 : \psi_{ij} = 0,\ i = 1, \cdots, I-1,\ \ j = 1, \cdots, J-1$, and $H_a : \psi_{ij} \geq 0$, with strict inequality for at least one pair (i, j).

Also let $T_{ij} = \Sigma_{l=1}^{j}\Sigma_{k=1}^{i} n_{kl}$, $\boldsymbol{T}_i = (T_{i1}, \cdots, T_{i(J-1)})$, $i = 1, \cdots, I-1$, and $\boldsymbol{T} = (\boldsymbol{T}_1, \cdots, \boldsymbol{T}_{I-1})$. Attention can be restricted to the sufficient statistics $(\boldsymbol{T}, \boldsymbol{m})$ which have the joint distribution

$$f(\boldsymbol{t}, \boldsymbol{m}) = \beta(\boldsymbol{\psi}, \boldsymbol{b}, \boldsymbol{d}) \exp(\Sigma_i^{I-1}\Sigma_j^{J-1}\psi_{ij}t_{ij} + \Sigma_i^{I-1} n_{i+}b_i + \Sigma_j^{J-1} n_{+j}d_j) g(\boldsymbol{t}, \boldsymbol{m}). \quad (4.2)$$

Note that $(\boldsymbol{T}, \boldsymbol{m})$ is a one-to-one linear transformation from the space $\boldsymbol{N}$. Let us next consider the structure of an exact test. If one wishes an exact test such that the size is equal to nominal value, any test procedure would require possible randomizations on some points of the distribution of test statistic. For an observed table $\boldsymbol{N}$, a test chooses rejection or acceptance with certain probabilities that depend on $\boldsymbol{N}$, denoted by $\varphi(\boldsymbol{N})$ and $1 - \varphi(\boldsymbol{N})$, respectively. A randomized test is therefore completely characterized by φ, the critical function, with $0 \leq \varphi(\boldsymbol{N}) \leq 1$ for all $\boldsymbol{N}$. If $\varphi(\boldsymbol{N})$ takes on only the values 1 and 0, then this becomes a nonrandomized test. Let $\varphi(\boldsymbol{N})$ denote an exact test of size α depending on $\boldsymbol{T}$ and $\boldsymbol{m}$ for the hypotheses concerning the distribution of $\boldsymbol{N}$ (or the joint distribution of $\boldsymbol{T}$ and $\boldsymbol{m}$), and also

denote the conditional test as a function of $\boldsymbol{t}$, for each fixed $\boldsymbol{m}$, by $\varphi_{\boldsymbol{m}}(\boldsymbol{t})$. If the conditional test has conditional size α, then the size of the original test can be obtained from the conditional tests by taking the expectation over $\boldsymbol{m}$, which is $E_{\boldsymbol{\psi},\boldsymbol{\nu}}\varphi(\boldsymbol{N}) = E_{\boldsymbol{\psi},\boldsymbol{\nu}}[E_{\boldsymbol{\psi}}(\varphi_{\boldsymbol{m}}(\boldsymbol{t})|\boldsymbol{m})] = \alpha$, where $\boldsymbol{\nu}$ refers to the nuisance parameters for $\boldsymbol{m}$.

By Lehmann (1986), $\boldsymbol{m}$ is sufficient and complete under the null, and any similar test of size α must have Neyman structure. Hence, the test for each fixed $\boldsymbol{m}$, $\varphi_{\boldsymbol{m}}(\boldsymbol{t})$, must have conditional size α, i.e., $E_{\boldsymbol{\psi}=\boldsymbol{0}}[\varphi_{\boldsymbol{m}}(\boldsymbol{t})|\boldsymbol{m}] = \alpha$ for all $\boldsymbol{m}$. Accordingly, if $\varphi(\boldsymbol{N})$ is size α, then $\varphi_{\boldsymbol{m}}(\boldsymbol{t})$ is of size α for each fixed $\boldsymbol{m}$. The conditional distribution of $\boldsymbol{T}$ can be obtained by fixing $\boldsymbol{m}$, which is free from nuisance parameters, and the test $\varphi_{\boldsymbol{m}}(\boldsymbol{t})$ is done conditionally given the values of the sufficient statistics, $\boldsymbol{m}$, for the nuisance parameters under the null. Hence, the critical values depend on these values.

We want to establish conditions under which the overall test is unbiased and admissible. Suppose for each $\boldsymbol{m}$, $\varphi_{\boldsymbol{m}}(\boldsymbol{t})$ is monotone nondecreasing in $\boldsymbol{t}$. This means that when all elements of $\boldsymbol{t}$ are fixed except for any one, $\varphi_{\boldsymbol{m}}(\boldsymbol{t})$ is nondecreasing in that variable. Next, we let for each fixed $\boldsymbol{m}$, $A_{\varphi_{\boldsymbol{m}}}(\boldsymbol{t}) = \{\boldsymbol{t} : \varphi_{\boldsymbol{m}}(\boldsymbol{t}) < 1\}$. Hence, $A_{\varphi_{\boldsymbol{m}}}(\boldsymbol{t})$ is the acceptance region of the test, except for possible randomization. A point $a \in A$ is called an extreme point if a is not an interior point of any line segment in A. Cohen and Sackrowitz (1991) gave the class of tests that are simultaneously exact, unbiased, and admissible.

<u>Theorem 4.2.1</u> For each fixed $\boldsymbol{m}$, if $\varphi_{\boldsymbol{m}}(\boldsymbol{t})$ is monotone nondecreasing in $\boldsymbol{t}$, then the test $\varphi_{\boldsymbol{m}}(\boldsymbol{t})$ is conditionally unbiased and the original test $\varphi(\boldsymbol{N})$ is unconditionally unbiased. Furthermore, the test $\varphi(\boldsymbol{N})$ is admissible if and only if for each fixed $\boldsymbol{m}$, $A_{\varphi_{\boldsymbol{m}}}(\boldsymbol{t})$ is convex and $\varphi_{\boldsymbol{m}}(\boldsymbol{t})$ is zero at nonextreme points of $A_{\varphi_{\boldsymbol{m}}}$.

Hence, an exact test is unbiased and admissible if and only if conditionally, given $\boldsymbol{m}$, the acceptance regions are monotone (in the sense that the corresponding $\varphi_{\boldsymbol{m}}(\boldsymbol{t})$ is monotone) and convex with randomization possible only at extreme points.

The following definitions and lemmas are used for the proof of the unbiasedness of tests in Theorem 4.2.1. Let $\boldsymbol{x}$ be a $k \times 1$ vector lying in $\boldsymbol{X}^k = X_1 \times X_2 \times \cdots \times X_k$, where X_i is a totally ordered subset of $\boldsymbol{R}^1$. Let $\boldsymbol{H}_k$ denote the family of nondecreasing functions on $\boldsymbol{X}^k$. Let $\boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{X}^k$, and $f(\boldsymbol{x})$ be a nonnegative function defined on $\boldsymbol{X}^k$ satisfying

$$f(\boldsymbol{x} \vee \boldsymbol{y}) f(\boldsymbol{x} \wedge \boldsymbol{y}) \geq f(\boldsymbol{x}) f(\boldsymbol{y}), \tag{4.3}$$

where $\vee$ and $\wedge$ are the corresponding lattice operations on $\boldsymbol{X}^k$, i.e., for $\boldsymbol{x} = (x_1, \cdots, x_k)$, $\boldsymbol{y} = (y_1, \cdots, y_k)$

$$\boldsymbol{x} \vee \boldsymbol{y} = (\max(x_1, y_1), \max(x_2, y_2), \cdots, \max(x_k, y_k))$$

and

$$\boldsymbol{x} \wedge \boldsymbol{y} = (\min(x_1, y_1), \min(x_2, y_2), \cdots, \min(x_k, y_k)).$$

From Karlin and Rinott (1980) we have the following definition.

Definition 4.2.1 A function with the property (4.3) is said to be multivariate totally positive of order 2 (MTP_2) on $\boldsymbol{X}^k$. Also a $k \times 1$ random vector $\boldsymbol{U} = (U_1, \cdots, U_k)$ is MTP_2 if its density is MTP_2 .

The multivariate total positivity is defined in terms of ordering on a lattice. Karlin and Rinott showed that if $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ are MTP_2 on $\boldsymbol{X}^k$, then $f(\boldsymbol{x})g(\boldsymbol{x})$ is MTP_2 on $\boldsymbol{X}^k$. Also, if $f(\boldsymbol{x}) = g(x_i, x_j)$, where g is TP_2 on $X_i \times X_j$, then f is MTP_2 on $\boldsymbol{X}^k$. Hence, products of such functions are MTP_2 on $\boldsymbol{X}^k$. With connection to MTP_2 density, Fortuin, Ginibre, and Kasteleyn (1971) stated the following inequality, which we denote by FGK. Let $\boldsymbol{U}$ be a random vector whose density is MTP_2, with respect

to a product measure defined on a product set. Let W_1, W_2 be functions of $\boldsymbol{U}$ lying in $\boldsymbol{H}_k$. Then

$$E(W_1, W_2) \geq EW_1 EW_2. \tag{4.4}$$

Now let $\boldsymbol{u} \leq \boldsymbol{v}$ mean $u_i \leq v_i, \quad i = 1, 2, \cdots, k$. From Marshall and Olkin (1970) we have the following definition.

Definition 4.2.2 A random vector $\boldsymbol{U}$ is said to be stochastically less than or equal to a random vector $\boldsymbol{V}$ if

$$Eh(\boldsymbol{U}) \leq^p Eh(\boldsymbol{V}), \tag{4.5}$$

for all $h \in \boldsymbol{H}_k$ for which expectations exist.

These definitions and inequality were incorporated in the following lemmas. These lemmas were provided by Cohen and Sackrowitz (1991), and applied to show the unbiasedness of tests.

Lemma 4.2.1 Assume H_0 is true. Also assume, conditional on $\boldsymbol{m} = (\{n_{i+}\}, \{n_{+j}\})$, $i = 1, \cdots, I-1, \quad j = 1, \cdots, J-1$,

$$\boldsymbol{T}_1 \text{ is } MTP_2 \tag{4.6}$$

$$\boldsymbol{T}_i | \boldsymbol{T}_1, \cdots, \boldsymbol{T}_{i-1} \text{ is } MTP_2 \text{ for all } i = 2, 3, \cdots, I-1 \tag{4.7}$$

$$\begin{aligned} \boldsymbol{T}_i | \boldsymbol{T}_1, \cdots, \boldsymbol{T}_{i-1} \quad \leq^p \quad & \boldsymbol{T}_i | \boldsymbol{T}'_1, \cdots, \boldsymbol{T}'_{i-1} \text{ for all } i = 2, 3, \cdots, I-1 \\ if \quad & \boldsymbol{T}_j \leq \boldsymbol{T}'_j \text{ for } j = 1, \cdots, i-1. \end{aligned} \tag{4.8}$$

Let $W(\boldsymbol{T}_1, \cdots, \boldsymbol{T}_{I-1})$ and $W^*(\boldsymbol{T}_1, \cdots, \boldsymbol{T}_{I-1}) \in \boldsymbol{H}_{(I-1)(J-1)}$. Then under H_0,

$$E\{W(\boldsymbol{T})W^*(\boldsymbol{T})|\boldsymbol{m}\} \geq E\{W(\boldsymbol{T})|\boldsymbol{m}\}E\{W^*(\boldsymbol{T})|\boldsymbol{m}\}. \tag{4.9}$$

The next lemmas are three conditions assumed for the previous lemma, and the proofs are given by Cohen and Sackrowitz (1991).

Lemma 4.2.2 Under $H_0, \boldsymbol{T}_1$ given $\boldsymbol{m}$ is MTP_2.

Lemma 4.2.3 Under $H_0, \boldsymbol{T}_i|\boldsymbol{T}_1, \cdots, \boldsymbol{T}_{i-1}, \boldsymbol{m}$ is MTP_2 for all $i = 2, \cdots, I-1$.

Lemma 4.2.4 Under $H_0, \boldsymbol{T}_i|\boldsymbol{T}_1, \cdots, \boldsymbol{T}_{i-1}, \boldsymbol{m} \leq^p \boldsymbol{T}_i|\boldsymbol{T}'_1, \cdots, \boldsymbol{T}'_{i-1}, \boldsymbol{m}$ for all $i = 2, 3, \cdots, I-1$ if $\boldsymbol{T}_j \leq \boldsymbol{T}'_j$ for $j = 1, \cdots, i-1$.

Cohen and Sackrowitz proved the unbiasedness portion of Theorem 4.2.1 for two-way tables. These lemmas and inequality are the main tools for the proof. Now, we want to display a test statistic for two-way tables and show that tests based on it have a desirable monotonicity property.

For a two-way contingency table where each classification is ordered, the statistic to reflect the association between two ordinal variables is

$$T = \Sigma\Sigma u_i v_j n_{ij}, \tag{4.10}$$

where u_i's and v_j's are monotone scores to display category ordering. This statistic is studied in Agresti (1990, 1992). If $u_i = (I - (i-1))$ and $\nu_j = (J - (j-1))$, then it becomes

$$T = \Sigma_{j=1}^{J} \Sigma_{i=1}^{I} t_{ij}, \tag{4.11}$$

where $t_{ij} = \Sigma_{l=1}^{j} \Sigma_{m=1}^{i} n_{ml}$. Next, we show that tests based on $T = \Sigma\Sigma t_{ij}$ have a desirable monotonicity property; hence, they are unbiased. We note that by the definition of monotonicity (all t_{ij} are fixed, except one) for any I, J the statistic $T = \Sigma\Sigma t_{ij}$ is monotone in t_{ij}, $i = 1, \cdots, I-1$, $j = 1, \cdots, J-1$. Hence, the test based on T is monotone and then unbiased, since it satisfies the condition of Theorem 4.2.1. This is the test statistic that Cohen and Sackrowitz used in two-way tables with ordinal alternative.

For the admissibility portion, Ledwina's theorem will be applied and stated in Section 4.5. The unbiased and admissible tests will be explained in Section 4.6, and we will show that the tests satisfy the properties of Theorem 4.2.1.

4.3 Unbiasedness of Tests in Three-way Contingency Tables

In this section, we will generalize Theorem 4.2.1 for testing conditional independence in three-way contingency tables. The unbiasedness portion of the tests in Theorem 4.2.1 considering three-way tables is proved with lemmas, and we utilize the definitions and lemmas stated in Section 4.2. For the admissibility part, we will apply the theorems in Ledwina (1978a, 1984) and Matthes and Truax (1967) for exponential families, which will be stated in the next sections. Showing unbiasedness of tests is the main part for proving Theorem 4.2.1 in three-way tables, and we will follow the arguments in Ledwina for the admissibility of the tests. Then we have the exact, unbiased, and admissible tests in three-way tables.

4.3.1 Conditional Independence Model

We will specify the general multinomial model in three-way tables, and state the testing problem under the null hypothesis of conditional independence. We will prove unbiasedness of tests and related lemmas focusing on three-way tables. Consider an $I \times J \times K$ contingency table under the multinomial model, where each row and column classification is ordinal. Let $\boldsymbol{N} = \{n_{ijk}\}$ denote observed cell counts, with expected frequencies $\{m_{ijk}\}$. Let $\boldsymbol{\pi} = \{\pi_{ijk}\}$ be probabilities for a multinomial distribution

over $I \times J \times K$ cells, where $n = \Sigma\Sigma\Sigma n_{ijk}$, and $\Sigma\Sigma\Sigma\pi_{ijk} = 1$. From Ledwina (1978b), the distribution of $\boldsymbol{N}$ can be written as

$$\begin{aligned}
f(\boldsymbol{N}) &= \frac{n!}{\Pi_{i,j,k}^{I,J,K} n_{ijk}!} \Pi_{i,j,k}^{I,J,K} \pi_{ijk}^{n_{ijk}} \\
&= \frac{l^n n!}{\Pi_{i,j,k}^{I,J,K} n_{ijk}!} \exp(\Sigma_i^{I-1}\Sigma_j^{J-1}\Sigma_k^{K-1} n_{ijk}a_{ijk} + \Sigma_i^{I-1}\Sigma_j^{J-1} n_{ij+}b_{ij} \\
&+ \Sigma_i^{I-1}\Sigma_k^{K-1} n_{i+k}c_{ik} + \Sigma_j^{J-1}\Sigma_k^{K-1} n_{+jk}d_{jk} + \Sigma_{i=1}^{I-1} n_{i++}e_i \\
&+ \Sigma_{j=1}^{J-1} n_{+j+}t_j + \Sigma_{k=1}^{K-1} n_{++k}g_k), \qquad (4.12)
\end{aligned}$$

where

$$\begin{aligned}
a_{ijk} &= \log \frac{\pi_{ijk}\pi_{IJk}}{\pi_{iJk}\pi_{Ijk}} \cdot \frac{\pi_{iJK}\pi_{IjK}}{\pi_{ijK}\pi_{IJK}} = \log \frac{\pi_{ijk}\pi_{IJk}}{\pi_{iJk}\pi_{Ijk}} - \log \frac{\pi_{ijK}\pi_{IJK}}{\pi_{iJK}\pi_{IjK}} \\
b_{ij} &= \log \frac{\pi_{ijK}\pi_{IJK}}{\pi_{iJK}\pi_{IjK}} \\
c_{ik} &= \log \frac{\pi_{iJk}\pi_{IJK}}{\pi_{iJK}\pi_{IJk}} \\
d_{jk} &= \log \frac{\pi_{Ijk}\pi_{IJK}}{\pi_{IjK}\pi_{IJk}} \\
e_i &= \log \frac{\pi_{iJK}}{\pi_{IJK}}, \quad f_j = \log \frac{\pi_{IjK}}{\pi_{IJK}}, \quad g_k = \log \frac{\pi_{IJk}}{\pi_{IJK}},
\end{aligned}$$

and

$$l = (1+\Sigma e^{e_i}+\Sigma e^{f_j}+\Sigma e^{g_k}+\Sigma e^{e_i+f_j+b_{ij}}+\Sigma e^{e_i+g_k+c_{ik}}+\Sigma e^{f_j+g_k+d_{jk}}+\Sigma e^{e_i+f_j+g_k+b_{ij}+c_{ik}+d_{jk}+a_{ijk}})^{-1}.$$

Let

$$\psi_{ij(k)} = \log \frac{\pi_{ijk}\pi_{i+1,j+1,k}}{\pi_{i,j+1,k}\pi_{i+1,j,k}},$$

which is the local log odds ratios in kth stratum. Note that

$$\log \frac{\pi_{lmk}\pi_{IJk}}{\pi_{lJk}\pi_{Imk}} = \Sigma_{i=l}^{I-1}\Sigma_{j=m}^{J-1}\psi_{ij(k)}.$$

Hence, we have

$$a_{lmk} = \Sigma_{i=l}^{I-1}\Sigma_{j=m}^{J-1}(\psi_{ij(k)} - \psi_{ij(K)}).$$

Also let $T_{ij(k)} = \Sigma_{m=1}^{j}\Sigma_{l=1}^{i}n_{lmk}$, and $\boldsymbol{T}_i^{(k)} = (T_{i1(k)}, T_{i2(k)}, \cdots, T_{i(J-1)(k)})$, $i = 1, \cdots, I-1$, and $\boldsymbol{T} = (\boldsymbol{T}_1^{(1)}, \cdots, \boldsymbol{T}_{I-1}^{(1)}, \boldsymbol{T}_1^{(2)}, \cdots, \boldsymbol{T}_{I-1}^{(2)}, \cdots, \boldsymbol{T}_1^{(K)}, \cdots, \boldsymbol{T}_{I-1}^{(K)})$. Then

$$\begin{aligned}
\Sigma_{k=1}^{K-1}[\Sigma_{i=1}^{I-1}\Sigma_{j=1}^{J-1}(\psi_{ij(k)} - \psi_{ij(K)})T_{ij(k)}] &= \Sigma_{k=1}^{K-1}[\Sigma_{i=1}^{I-1}\Sigma_{j=1}^{J-1}(\psi_{ij(k)} - \psi_{ij(K)})\Sigma_{m=1}^{j}\Sigma_{l=1}^{i}n_{lmk}] \\
&= \Sigma_{k=1}^{K-1}[\Sigma_{i=1}^{I-1}\Sigma_{j=1}^{J-1}\Sigma_{m=1}^{j}\Sigma_{l=1}^{i}(\psi_{ij(k)} - \psi_{ij(K)})n_{lmk}] \\
&= \Sigma_{k=1}^{K-1}\Sigma_{l=1}^{I-1}\Sigma_{m=1}^{J-1}(\Sigma_{i=l}^{I-1}\Sigma_{j=m}^{J-1}(\psi_{ij(k)} - \psi_{ij(K)}))n_{lmk}] \\
&= \Sigma_{k=1}^{K-1}[\Sigma_{i=1}^{I-1}\Sigma_{j=1}^{J-1}a_{ijk}n_{ijk}]. \qquad (4.13)
\end{aligned}$$

Let $\boldsymbol{r} = (\{n_{ij+}\}, \{n_{i+k}\}, \{n_{+jk}\})$. Then using (4.13) we rewrite (4.12) as

$$\begin{aligned}
f(\boldsymbol{N}) &= \beta(\boldsymbol{\psi}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d}, \boldsymbol{e}, \boldsymbol{f}, \boldsymbol{g})\exp(\Sigma_k^{K-1}\Sigma_i^{I-1}\Sigma_j^{J-1}(\psi_{ij(k)} - \psi_{ij(K)})T_{ij(k)} \\
&+ \Sigma_i^{I-1}\Sigma_j^{J-1}n_{ij+}b_{ij} + \Sigma_i^{I-1}\Sigma_k^{K-1}n_{i+k}c_{ik} + \Sigma_j^{J-1}\Sigma_k^{K-1}n_{+jk}d_{jk} \\
&+ \Sigma_{i=1}^{I-1}n_{i++}e_i + \Sigma_{j=1}^{J-1}n_{+j+}f_j + \Sigma_{k=1}^{K-1}n_{++k}g_k) \cdot g(\boldsymbol{t}, \boldsymbol{r}). \qquad (4.14)
\end{aligned}$$

Hence, no three-factor interaction has the following equivalent expressions, for all i and j:

$$a_{ijk} = 0, \quad k = 1, \cdots, K-1$$

$$\log\frac{\pi_{ijk}\pi_{IJk}}{\pi_{iJk}\pi_{Ijk}} - \log\frac{\pi_{ijK}\pi_{IJK}}{\pi_{iJK}\pi_{IjK}} = 0, \quad k = 1, \cdots, K-1$$

$$\psi_{ij(k)} = \psi_{ij(K)}, \ k = 1, \cdots, K-1$$

$$\psi_{ij(1)} = \psi_{ij(2)} = \cdots = \psi_{ij(K)} = \psi_{ij}.$$

It means that the association between row and column variables is identical at each level of stratum.

When we test the model of conditional independence, we will assume that the model of no three-factor interaction holds. Hence, we assume that for all i and j, $(\psi_{ij(k)} - \psi_{ij(K)}) = 0$, $k = 1, \cdots, K-1$. Let $T_{ij+} = \Sigma_{m=1}^{j}\Sigma_{l=1}^{i} n_{lm+}$. Then

$$\begin{aligned}
\Sigma_{i=1}^{I-1}\Sigma_{j=1}^{J-1}\psi_{ij(K)}T_{ij+} &= \Sigma_{i=1}^{I-1}\Sigma_{j=1}^{J-1}\psi_{ij(K)}(\Sigma_{m=1}^{j}\Sigma_{l=1}^{i} n_{lm+}) \\
&= \Sigma_{i=1}^{I-1}\Sigma_{j=1}^{J-1}\Sigma_{m=1}^{j}\Sigma_{l=1}^{i}\psi_{ij(K)}n_{lm+} \\
&= \Sigma_{l=1}^{I-1}\Sigma_{m=1}^{J-1}(\Sigma_{i=l}^{I-1}\Sigma_{j=m}^{J-1}\psi_{ij(K)})n_{lm+} \\
&= \Sigma_{i=1}^{I-1}\Sigma_{j=1}^{J-1} b_{ij}n_{ij+}. \qquad (4.15)
\end{aligned}$$

Let $\boldsymbol{m} = (\{n_{i+k}\}, \{n_{+jk}\})$, $i = 1, \cdots, I-1$, $j = 1, \cdots, J-1$, $k = 1, \cdots, K$. Using (4.15), we rewrite (4.14) as

$$\begin{aligned}
f(\boldsymbol{N}) &= \beta(\boldsymbol{\psi}, \boldsymbol{c}, \boldsymbol{d}, \boldsymbol{e}, \boldsymbol{f}, \boldsymbol{g}) \exp(\Sigma_{k}^{K-1}\Sigma_{i}^{I-1}\Sigma_{j}^{J-1}(\psi_{ij(k)} - \psi_{ij(K)})T_{ij(k)} \\
&+ \Sigma_{i}^{I-1}\Sigma_{j}^{J-1}\psi_{ij(K)}T_{ij+} + \Sigma_{i}^{I-1}\Sigma_{k}^{K-1} n_{i+k}c_{ik} + \Sigma_{j}^{J-1}\Sigma_{k}^{K-1} n_{+jk}d_{jk} \\
&+ \Sigma_{i=1}^{I-1} n_{i++}e_i + \Sigma_{j=1}^{J-1} n_{+j+}f_j + \Sigma_{k=1}^{K-1} n_{++k}g_k) \cdot g(\boldsymbol{t}, \boldsymbol{m}). \qquad (4.16)
\end{aligned}$$

Note that $T_{ij+} = \Sigma_{m=1}^{j}\Sigma_{l=1}^{i} n_{lm+} = \Sigma_{k=1}^{K} T_{ij(k)}$. Then

$$\begin{aligned}
&\Sigma_{k}^{K-1}\Sigma_{i}^{I-1}\Sigma_{j}^{J-1}(\psi_{ij(k)} - \psi_{ij(K)})T_{ij(k)} + \Sigma_{i=1}^{I-1}\Sigma_{j=1}^{J-1}\psi_{ij(K)}T_{ij+} \\
&= \Sigma_{k}^{K-1}\Sigma_{i}^{I-1}\Sigma_{j}^{J-1}(\psi_{ij(k)} - \psi_{ij(K)})T_{ij(k)} + \Sigma_{i}^{I-1}\Sigma_{j}^{J-1}\psi_{ij(K)}\Sigma_{k=1}^{K} T_{ij(k)} \\
&= \Sigma_{k}^{K-1}\Sigma_{i}^{I-1}\Sigma_{j}^{J-1}\psi_{ij(k)}T_{ij(k)} - \Sigma_{k}^{K-1}\Sigma_{i}^{I-1}\Sigma_{j}^{J-1}\psi_{ij(K)}T_{ij(k)} \\
&+ \Sigma_{k=1}^{K}\Sigma_{i}^{I-1}\Sigma_{j}^{J-1}\psi_{ij(K)}T_{ij(k)} \\
&= \Sigma_{k}^{K-1}\Sigma_{i}^{I-1}\Sigma_{j}^{J-1}\psi_{ij(k)}T_{ij(k)} + \Sigma_{i}^{I-1}\Sigma_{j}^{J-1}\psi_{ij(K)}T_{ij(K)} \\
&= \Sigma_{k=1}^{K}\Sigma_{i}^{I-1}\Sigma_{j}^{J-1}\psi_{ij(k)}T_{ij(k)}. \qquad (4.17)
\end{aligned}$$

Hence, using (4.17), we rewrite (4.16) as

$$\begin{aligned} f(\boldsymbol{N}) &= \beta(\boldsymbol{\psi}, \boldsymbol{c}, \boldsymbol{d}, \boldsymbol{e}, \boldsymbol{f}, \boldsymbol{g}) \exp(\Sigma_k^K \Sigma_i^{I-1} \Sigma_j^{J-1} \psi_{ij(k)} T_{ij(k)} \\ &+ \Sigma_i^{I-1} \Sigma_k^{K-1} n_{i+k} c_{ik} + \Sigma_j^{J-1} \Sigma_k^{K-1} n_{+jk} d_{jk} \\ &+ \Sigma_{i=1}^{I-1} n_{i++} e_i + \Sigma_{j=1}^{J-1} n_{+j+} f_j + \Sigma_{k=1}^{K-1} n_{++k} g_k) \cdot g(\boldsymbol{t}, \boldsymbol{m}). \end{aligned} \tag{4.18}$$

From (4.18), we see that we may treat the observation as $t_{ij(k)}$ and the parameters as $\psi_{ij(k)}$. The problem is to test conditional independence under the assumption that the model of no three-factor interaction holds. Here we consider the problem under the simplifying assumption that the $\psi_{ij(k)}$ have a common ψ over k, so that the hypothesis reduces to $H_0 : \psi = 0$, when the ψ's are not assumed to be equal but $\psi_{ij(k)} = \psi_{ij}, k = 1, \cdots, K$ for $i = 1, \cdots, I-1,\ j = 1, \cdots, J-1$. Therefore, our hypotheses become

$$\begin{aligned} H_0 &: \psi_{ij(k)} = \psi_{ij(K)} \text{ and } \psi_{ij(K)} = 0,\ \ i = 1, \cdots, I-1,\ \ j = 1, \cdots, J-1,\ \ k = 1, \cdots, K-1 \\ &\Leftrightarrow \psi_{ij(k)} = \psi = 0,\ \ i = 1, \cdots, I-1,\ \ j = 1, \cdots, J-1,\ \ k = 1, \cdots, K \\ H_a &: \text{No Three-Factor Interaction Model.} \end{aligned}$$

The test is carried out conditionally, given the values of margins, and the conditional joint distribution of $\boldsymbol{N}$ given $\boldsymbol{m}$ under the null reduces to the product of K hypergeometric mass functions, which is the table probability under the null.

4.3.2 Unbiasedness of Tests

In order to prove unbiasedness of tests in Theorem 4.2.1, we need the following lemma. In the lemma, three conditions are assumed and they will be verified after proving unbiasedness in Theorem 4.2.1. This test is done by conditioning on the

values of all elements in the margins, $\boldsymbol{m} = (\{n_{i+k}\}, \{n_{+jk}\})$, that are random, so that in the conditional model these margins $\boldsymbol{m}$ are fixed, and cell counts from different strata are independent.

Lemma 4.3.1 Assume H_0 is true. Also assume, conditional on $\boldsymbol{m}$,

$$\text{i) } \boldsymbol{T}_1^{(k)} \text{ is } MTP_2 \text{ for all } k = 1, \cdots, K \tag{4.19}$$

$$\text{ii) } \boldsymbol{T}_i^{(k)} | \boldsymbol{T}_1^{(1)}, \cdots, \boldsymbol{T}_{i-1}^{(1)}, \boldsymbol{T}_1^{(2)}, \cdots, \boldsymbol{T}_{i-1}^{(2)}, \cdots, \boldsymbol{T}_1^{(K)}, \cdots, \boldsymbol{T}_{i-1}^{(K)} \text{ is } MTP_2$$
$$\text{for all } i = 2, 3, \cdots, I-1, \;\; k = 1, \cdots, K \tag{4.20}$$

$$\text{iii) } \boldsymbol{T}_i^{(k)} | \boldsymbol{T}_1^{(1)}, \cdots, \boldsymbol{T}_{i-1}^{(1)}, \boldsymbol{T}_1^{(2)}, \cdots, \boldsymbol{T}_{i-1}^{(K)} \leq^p \boldsymbol{T}_i^{(k)} | \boldsymbol{T'}_1^{(1)}, \cdots, \boldsymbol{T'}_{i-1}^{(1)}, \boldsymbol{T'}_1^{(2)}, \cdots, \boldsymbol{T'}_{i-1}^{(K)}$$
$$\text{for all } i = 2, 3, \cdots, I-1, \;\; k = 1, \cdots, K,$$

$$\text{if } \boldsymbol{T}_j^{(k)} \leq \boldsymbol{T'}_j^{(k)} \text{for } j = 1, \cdots, i-1, \;\; k = 1, \cdots, K. \tag{4.21}$$

Let $W(\boldsymbol{T}) = W(\boldsymbol{T}_1^{(1)}, \cdots, \boldsymbol{T}_{I-1}^{(1)}, \boldsymbol{T}_1^{(2)}, \cdots, \boldsymbol{T}_{I-1}^{(2)}, \cdots, \boldsymbol{T}_{I-1}^{(K)})$, and
$W^*(\boldsymbol{T}) = W^*(\boldsymbol{T}_1^{(1)}, \cdots, \boldsymbol{T}_{I-1}^{(1)}, \boldsymbol{T}_1^{(2)}, \cdots, \boldsymbol{T}_{I-1}^{(2)}, \cdots, \boldsymbol{T}_{I-1}^{(K)}) \in \boldsymbol{H}_{(K)(I-1)(J-1)}$, where $\boldsymbol{H}_{(K)(I-1)(J-1)}$ denote the family of nondecreasing functions on $\boldsymbol{R}^{(K)(I-1)(J-1)}$.
Then under H_0,

$$E\{W(\boldsymbol{T})W^*(\boldsymbol{T})|\boldsymbol{m}\} \geq E\{W(\boldsymbol{T})|\boldsymbol{m}\}E\{W^*(\boldsymbol{T})|\boldsymbol{m}\}. \tag{4.22}$$

Proof. We suppress $\boldsymbol{m}$, since all statements are conditional on $\boldsymbol{m}$. Now

$$\begin{aligned} EW(\boldsymbol{T})W^*(\boldsymbol{T}) &= EW(\boldsymbol{T}_1^{(1)}, \cdots, \boldsymbol{T}_{I-1}^{(1)}, \boldsymbol{T}_1^{(2)}, \cdots, \boldsymbol{T}_{I-1}^{(K)}) \\ &\quad W^*(\boldsymbol{T}_1^{(1)}, \cdots, \boldsymbol{T}_{I-1}^{(1)}, \boldsymbol{T}_1^{(2)}, \cdots, \boldsymbol{T}_{I-1}^{(K)}) \\ &= E\{EW(\boldsymbol{T}_1^{(1)}, \cdots, \boldsymbol{T}_{I-1}^{(K)})W^*(\boldsymbol{T}_1^{(1)}, \cdots, \boldsymbol{T}_{I-1}^{(K)})| \\ &\quad \boldsymbol{T}_1^{(1)}, \cdots, \boldsymbol{T}_{I-2}^{(1)}, \boldsymbol{T}_1^{(2)}, \cdots, \boldsymbol{T}_{I-2}^{(K)}\} \\ &\geq E\{E(W(\boldsymbol{T}_1^{(1)}, \cdots, \boldsymbol{T}_{I-1}^{(K)}|\boldsymbol{T}_1^{(1)}, \cdots, \boldsymbol{T}_{I-2}^{(K)}) \\ &\quad E(W^*(\boldsymbol{T}_1^{(1)}, \cdots, \boldsymbol{T}_{I-1}^{(K)})|\boldsymbol{T}_1^{(1)}, \cdots, \boldsymbol{T}_{I-2}^{(K)})\} \end{aligned} \tag{4.23}$$

by (4.20) and the FGK (Fortuin, Ginibre, and Kasteleyn) inequality. The expression (4.23) can be written as

$$EW_1(\boldsymbol{T}_1^{(1)},\cdots,\boldsymbol{T}_{I-2}^{(1)},\boldsymbol{T}_1^{(2)},\cdots,\boldsymbol{T}_{I-2}^{(K)})W_1^*(\boldsymbol{T}_1^{(1)},\cdots,\boldsymbol{T}_{I-2}^{(1)},\boldsymbol{T}_1^{(2)},\cdots,\boldsymbol{T}_{I-2}^{(K)}) \quad (4.24)$$

where

$$W_1(\boldsymbol{T}_1^{(1)},\cdots,\boldsymbol{T}_{I-2}^{(1)},\boldsymbol{T}_1^{(2)},\cdots,\boldsymbol{T}_{I-2}^{(K)}) = EW(\boldsymbol{T}_1^{(1)},\cdots,\boldsymbol{T}_{I-1}^{(K)}|\boldsymbol{T}_1^{(1)},\cdots,\boldsymbol{T}_{I-2}^{(K)})$$

and

$$W_1^*(\boldsymbol{T}_1^{(1)},\cdots,\boldsymbol{T}_{I-2}^{(1)},\boldsymbol{T}_1^{(2)},\cdots,\boldsymbol{T}_{I-2}^{(K)}) = EW^*(\boldsymbol{T}_1^{(1)},\cdots,\boldsymbol{T}_{I-1}^{(K)}|\boldsymbol{T}_1^{(1)},\cdots,\boldsymbol{T}_{I-2}^{(K)}).$$

Note that (4.21) implies W_1 and $W_1^* \in \boldsymbol{H}_{(K)(I-2)(J-1)}$. Therefore, one can use (4.20) again. Hence,

$$\begin{aligned}
&EW_1(\boldsymbol{T}_1^{(1)},\cdots,\boldsymbol{T}_{I-2}^{(1)},\boldsymbol{T}_1^{(2)},\cdots,\boldsymbol{T}_{I-2}^{(K)})W_1^*(\boldsymbol{T}_1^{(1)},\cdots,\boldsymbol{T}_{I-2}^{(1)},\boldsymbol{T}_1^{(2)},\cdots,\boldsymbol{T}_{I-2}^{(K)}) \\
&= E\{EW_1(\boldsymbol{T}_1^{(1)},\cdots,\boldsymbol{T}_{I-2}^{(1)},\boldsymbol{T}_1^{(2)},\cdots,\boldsymbol{T}_{I-2}^{(K)})W_1^*(\boldsymbol{T}_1^{(1)},\cdots,\boldsymbol{T}_{I-2}^{(1)},\boldsymbol{T}_1^{(2)},\cdots,\boldsymbol{T}_{I-2}^{(K)})| \\
&\quad \boldsymbol{T}_1^{(1)},\cdots,\boldsymbol{T}_{I-3}^{(1)},\boldsymbol{T}_2^{(1)},\cdots,\boldsymbol{T}_{I-3}^{(K)})\} \\
&\geq E\{E(W_1(\boldsymbol{T}_1^{(1)},\cdots,\boldsymbol{T}_{I-2}^{(K)}|\boldsymbol{T}_1^{(1)},\cdots,\boldsymbol{T}_{I-3}^{(K)})E(W_1^*(\boldsymbol{T}_1^{(1)},\cdots,\boldsymbol{T}_{I-2}^{(K)}|\boldsymbol{T}_1^{(1)},\cdots,\boldsymbol{T}_{I-3}^{(K)}))\} \\
&= EW_2(\boldsymbol{T}_1^{(1)},\cdots,\boldsymbol{T}_{I-3}^{(K)})W_2^*(\boldsymbol{T}_1^{(1)},\cdots,\boldsymbol{T}_{I-3}^{(K)}) \qquad (4.25)
\end{aligned}$$

by letting

$$W_2(\boldsymbol{T}_1^{(1)},\cdots,\boldsymbol{T}_{I-3}^{(K)}) = EW_1(\boldsymbol{T}_1^{(1)},\cdots,\boldsymbol{T}_{I-2}^{(K)}|\boldsymbol{T}^{(1)},\cdots,\boldsymbol{T}_{I-3}^{(K)}),$$

and

$$W_2^*(\boldsymbol{T}_1^{(1)},\cdots,\boldsymbol{T}_{I-3}^{(K)}) = EW_1^*(\boldsymbol{T}_1^{(1)},\cdots,\boldsymbol{T}_{I-2}^{(K)}|\boldsymbol{T}^{(1)},\cdots,\boldsymbol{T}_{I-3}^{(K)}). \quad (4.26)$$

The process can be repeated until we have that (4.25) is greater than or equal to

$$\begin{aligned}
&EW_{I-2}(\boldsymbol{T}_1^{(1)},\cdots,\boldsymbol{T}_1^{(K)})W_{I-2}^*(\boldsymbol{T}_1^{(1)},\cdots,\boldsymbol{T}_1^{(K)}) \\
&\geq EW_{I-2}(\boldsymbol{T}_1^{(1)},\cdots,\boldsymbol{T}_1^{(K)})EW_{I-2}^*(\boldsymbol{T}_1^{(1)},\cdots,\boldsymbol{T}_1^{(K)}). \qquad (4.27)
\end{aligned}$$

The last step comes from (4.19) and FGK inequality. Also by the definition of W_{I-2},

$$
\begin{aligned}
& EW_{I-2}(\boldsymbol{T}_1^{(1)}, \cdots, \boldsymbol{T}_1^{(K)}) \\
&= E\{EW_{I-3}(\boldsymbol{T}_1^{(1)}, \boldsymbol{T}_2^{(1)}, \boldsymbol{T}_1^{(2)}, \boldsymbol{T}_2^{(2)}, \cdots, \boldsymbol{T}_2^{(K)} | \boldsymbol{T}_1^{(1)}, \cdots, \boldsymbol{T}_1^{(K)})\} \\
&= EW_{I-3}(\boldsymbol{T}_1^{(1)}, \boldsymbol{T}_2^{(1)}, \boldsymbol{T}_1^{(2)}, \boldsymbol{T}_2^{(2)}, \cdots, \boldsymbol{T}_2^{(K)}) \\
&= E\{EW_{I-4}(\boldsymbol{T}_1^{(1)}, \boldsymbol{T}_2^{(1)}, \boldsymbol{T}_3^{(1)}, \boldsymbol{T}_1^{(2)}, \cdots, \boldsymbol{T}_3^{(K)} | \boldsymbol{T}_1^{(1)}, \boldsymbol{T}_2^{(1)}, \boldsymbol{T}_1^{(2)}, \cdots, \boldsymbol{T}_2^{(K)})\} \\
&= \cdots = EW(\boldsymbol{T}_1^{(1)}, \cdots, \boldsymbol{T}_{I-1}^{(1)}, \boldsymbol{T}_2^{(1)}, \cdots, \boldsymbol{T}_{I-1}^{(K)}).
\end{aligned}
\tag{4.28}
$$

Similarly for W^*. Using (4.28) on the right-hand side of (4.27) we have

$$
\begin{aligned}
E\{W(\boldsymbol{T})W^*(\boldsymbol{T})|\boldsymbol{m}\} &= E\{W(\boldsymbol{T}_1^{(1)}, \cdots, \boldsymbol{T}_{I-1}^{(1)}, \boldsymbol{T}_1^{(2)}, \cdots, \boldsymbol{T}_{I-1}^{(K)}) \\
&\quad W^*(\boldsymbol{T}_1^{(1)}, \cdots, \boldsymbol{T}_{I-1}^{(1)}, \boldsymbol{T}_1^{(2)}, \cdots, \boldsymbol{T}_{I-1}^{(K)})|\boldsymbol{m}\} \\
&\geq E\{W(\boldsymbol{T}_1^{(1)}, \cdots, \boldsymbol{T}_{I-1}^{(1)}, \boldsymbol{T}_1^{(2)}, \cdots, \boldsymbol{T}_{I-1}^{(K)})|\boldsymbol{m}\} \\
&\quad E\{W^*(\boldsymbol{T}_1^{(1)}, \cdots, \boldsymbol{T}_{I-1}^{(1)}, \boldsymbol{T}_1^{(2)}, \cdots, \boldsymbol{T}_{I-1}^{(K)})|\boldsymbol{m}\} \\
&= E\{W(\boldsymbol{T})|\boldsymbol{m}\}E\{W^*(\boldsymbol{T})|\boldsymbol{m}\}.
\end{aligned}
\tag{4.29}
$$

<u>Proof of Unbiasedness in Theorem 4.2.1</u>

Now we show unbiasedness of tests in Theorem 4.2.1 in three-way tables. Let $f_{\boldsymbol{\psi}}(\boldsymbol{t}|\boldsymbol{m})$, $\boldsymbol{T} = (\boldsymbol{T}_1^{(1)}, \cdots, \boldsymbol{T}_{I-1}^{(1)}, \boldsymbol{T}_1^{(2)}, \cdots, \boldsymbol{T}_{I-1}^{(2)}, \cdots, \boldsymbol{T}_1^{(K)}, \cdots, \boldsymbol{T}_{I-1}^{(K)})$, denote the conditional density of $\boldsymbol{T}|\boldsymbol{m}$, where $\boldsymbol{\psi}$ lies in the alternative space and let $f_{\boldsymbol{0}}(\boldsymbol{t}|\boldsymbol{m})$ be the conditional density under the null. Using (4.18) we derive conditional densities.

Then,

$$W^*(\boldsymbol{t}) = \frac{f_{\boldsymbol{\psi}}(\boldsymbol{t}|\boldsymbol{m})}{f_{\boldsymbol{0}}(\boldsymbol{t}|\boldsymbol{m})}$$
$$\propto \exp(\Sigma_k^K \Sigma_i^{I-1} \Sigma_j^{J-1} \psi_{ij(k)} T_{ij(k)}). \quad (4.30)$$

Hence, $W^*(\boldsymbol{t})$ is monotone nondecreasing in $\boldsymbol{T}$, and $W^*(\boldsymbol{t}) \in \boldsymbol{H}_{(K)(I-1)(J-1)}$ for any $\boldsymbol{\psi}$ in the alternative space. Also, by the assumption, test $\varphi_{\boldsymbol{m}}(\boldsymbol{t}) \in \boldsymbol{H}_{(K)(I-1)(J-1)}$. Consider for $\boldsymbol{\psi}$ in the alternative space,

$$\begin{aligned} E_{\boldsymbol{\psi}}(\varphi_{\boldsymbol{m}}(\boldsymbol{t})|\boldsymbol{m}) &= \Sigma \varphi_{\boldsymbol{m}}(\boldsymbol{t}) f_{\boldsymbol{\psi}}(\boldsymbol{t}|\boldsymbol{m}) \\ &= \Sigma \varphi_{\boldsymbol{m}}(\boldsymbol{t}) W^*(\boldsymbol{t}) f_{\boldsymbol{0}}(\boldsymbol{t}|\boldsymbol{m}) \\ &\geq [\Sigma \varphi_{\boldsymbol{m}}(\boldsymbol{t}) f_{\boldsymbol{0}}(\boldsymbol{t}|\boldsymbol{m})][\Sigma W^*(\boldsymbol{t}) f_{\boldsymbol{0}}(\boldsymbol{t}|\boldsymbol{m})], \text{ by (4.22)} \\ &= E_{\boldsymbol{0}} \varphi_{\boldsymbol{m}}(\boldsymbol{t}) \\ &= \alpha. \end{aligned} \quad (4.31)$$

By the application of Lemma 4.3.1, we have inequality. Expression (4.31) implies conditional unbiasedness of $\varphi_{\boldsymbol{m}}(\boldsymbol{t})$, which in turn implies unbiasedness of the original test $\varphi(\boldsymbol{N})$, by noting that $E_{\boldsymbol{\psi},\boldsymbol{\nu}}\varphi(\boldsymbol{N}) = E_{\boldsymbol{\psi},\boldsymbol{\nu}}[E_{\boldsymbol{\psi}}(\varphi_{\boldsymbol{m}}(\boldsymbol{t})|\boldsymbol{m})] \geq \alpha$, where $\boldsymbol{\nu}$ refers to the nuisance parameters. Hence, we finish the unbiasedness portion of Theorem 4.2.1.

Proof of Lemmas

Now we verify (4.19), (4.20), and (4.21), which are conditions assumed for Lemma 4.3.1.

i) Under $H_0, \boldsymbol{T}_1^{(k)}$ given $\boldsymbol{m}$ is MTP_2, for all $k = 1, \cdots, K$.

Proof.

Let $1 \leq k \leq K$, and $i = 1, \cdots, I-1, \qquad j = 1, \cdots, J-1$.

$$
\begin{aligned}
\boldsymbol{T}_1^{(k)} | \boldsymbol{m} \;\; & \Leftrightarrow \;\; \boldsymbol{T}_1^{(k)} | \{n_{i+k'}\}, \{n_{+jk'}\} \;\; k' = 1, \cdots, K \\
& \Leftrightarrow \;\; \boldsymbol{T}_1^{(k)} | \{n_{i+k}\}, \{n_{+jk}\}, \{n_{i+k'}\}, \{n_{+jk'}\}, \text{ where } k' = 1, \cdots, k-1, k+1, \cdots, K \\
& \Leftrightarrow \;\; \boldsymbol{T}_1^{(k)} | \{n_{i+k}\}, \{n_{+jk}\} \qquad \text{since } \{n_{i+k'}\}, \{n_{+jk'}\} \\
& \qquad \text{are independent of } T_1^{(k)}, \{n_{i+k}\}, \{n_{+jk}\}, \qquad (4.32)
\end{aligned}
$$

which is MTP_2 by Lemma 4.2.2.

ii) Under H_0,

$$
\boldsymbol{T}_i^{(k)} | \boldsymbol{T}_1^{(1)}, \cdots, \boldsymbol{T}_{i-1}^{(1)}, \boldsymbol{T}_1^{(2)}, \cdots, \boldsymbol{T}_{i-1}^{(2)}, \cdots, \boldsymbol{T}_1^{(K)}, \cdots, \boldsymbol{T}_{i-1}^{(K)}, \boldsymbol{m} \text{ is } MTP_2
$$
$$
\text{for all } i = 2, 3, \cdots, I-1, \;\; k = 1, \cdots, K.
$$

Proof.

Let $1 \leq k \leq K$, and for all $i = 2, \cdots, I-1$,

$$
\begin{aligned}
& \boldsymbol{T}_i^{(k)} | \boldsymbol{T}_1^{(1)}, \cdots, \boldsymbol{T}_{i-1}^{(1)}, \boldsymbol{T}_1^{(2)}, \cdots, \boldsymbol{T}_{i-1}^{(2)}, \cdots, \boldsymbol{T}_1^{(K)}, \cdots, \boldsymbol{T}_{i-1}^{(K)}, \boldsymbol{m} \\
& \Leftrightarrow \;\; \boldsymbol{T}_i^{(k)} | \boldsymbol{T}_1^{(k)}, \cdots, \boldsymbol{T}_{i-1}^{(k)}, \boldsymbol{T}_1^{(1)}, \cdots, \boldsymbol{T}_{i-1}^{(1)}, \cdots, \boldsymbol{T}_1^{(k-1)}, \cdots, \boldsymbol{T}_{i-1}^{(k-1)}, \boldsymbol{T}_1^{(k+1)}, \cdots, \boldsymbol{T}_{i-1}^{(K)}, \boldsymbol{m} \\
& \Leftrightarrow \;\; \boldsymbol{T}_i^{(k)} | \boldsymbol{T}_1^{(k)}, \cdots, \boldsymbol{T}_{i-1}^{(k)}, \{n_{i+k}\}, \{n_{+jk}\} \qquad \text{by the independence of the strata,}
\end{aligned}
$$

which is MTP_2 by Lemma 4.2.3.

iii) Under H_0, given $\boldsymbol{m}$,

$$\boldsymbol{T}_i^{(k)}|\boldsymbol{T}_1^{(1)},\cdots,\boldsymbol{T}_{i-1}^{(1)},\boldsymbol{T}_1^{(2)},\cdots,\boldsymbol{T}_{i-1}^{(K)} \leq^p \boldsymbol{T}_i^{(k)}|\boldsymbol{T}'^{(1)}_1,\cdots,\boldsymbol{T}'^{(1)}_{i-1},\boldsymbol{T}'^{(2)}_1,\cdots,\boldsymbol{T}'^{(K)}_{i-1}$$

$$\text{for all } i=2,3,\cdots,I-1,\quad k=1,\cdots,K \quad (4.33)$$

$$\text{if } \boldsymbol{T}_j^{(k)} \leq \boldsymbol{T}'^{(k)}_j \text{for } j=1,\cdots,i-1,\quad k=1,\cdots,K.$$

Proof.

Let $1 \leq k \leq K$, and $i'=1,\cdots,I-1,\ \ j'=1,\cdots,J-1$.

For all $i=2,3,\cdots,I-1$,

$$\boldsymbol{T}_i^{(k)}|\boldsymbol{T}_1^{(1)},\cdots,\boldsymbol{T}_{i-1}^{(1)},\boldsymbol{T}_1^{(2)},\cdots,\boldsymbol{T}_{i-1}^{(K)},\boldsymbol{m}$$

$$\Leftrightarrow \quad \boldsymbol{T}_i^{(k)}|\boldsymbol{T}_1^{(k)},\cdots,\boldsymbol{T}_{i-1}^{(k)},\{n_{i'+k}\},\{n_{+j'k}\}, \qquad \text{by the independence of the strata.}$$

Likewise,

$$\boldsymbol{T}_i^{(k)}|\boldsymbol{T}'^{(1)}_1,\cdots,\boldsymbol{T}'^{(1)}_{i-1},\boldsymbol{T}'^{(2)}_1,\cdots,\boldsymbol{T}'^{(K)}_{i-1},\boldsymbol{m}$$

$$\Leftrightarrow \quad \boldsymbol{T}_i^{(k)}|\boldsymbol{T}'^{(k)}_1,\cdots,\boldsymbol{T}'^{(k)}_{i-1},\{n_{i'+k}\},\{n_{+j'k}\}, \qquad \text{by the independence of the strata.}$$

Hence,

$$\boldsymbol{T}_i^{(k)}|\boldsymbol{T}_1^{(1)},\cdots,\boldsymbol{T}_{i-1}^{(1)},\boldsymbol{T}_1^{(2)},\cdots,\boldsymbol{T}_{i-1}^{(K)} \leq^p \boldsymbol{T}_i^{(k)}|\boldsymbol{T}'^{(1)}_1,\cdots,\boldsymbol{T}'^{(1)}_{i-1},\boldsymbol{T}'^{(2)}_1,\cdots,\boldsymbol{T}'^{(K)}_{i-1}$$

$$\Leftrightarrow \quad \boldsymbol{T}_i^{(k)}|\boldsymbol{T}_1^{(k)},\cdots,\boldsymbol{T}_{i-1}^{(k)},\{n_{i'+k}\},\{n_{+j'k}\}, \leq^p \boldsymbol{T}_i^{(k)}|\boldsymbol{T}'^{(k)}_1,\cdots,\boldsymbol{T}'^{(k)}_{i-1},\{n_{i'+k}\},\{n_{+j'k}\},$$

$$\text{for all } i=2,3,\cdots,I-1$$

$$\text{if } \boldsymbol{T}_j^{(k)} \leq \boldsymbol{T}'^{(k)}_j \ \text{ for } j=1,\cdots,i-1,$$

which is proven by Lemma 4.2.4.

Hence, all three conditions assumed for Lemma 4.3.1 are established. We next present the complete class of tests and admissible tests in an exponential family.

4.4 Complete Class of Tests

We show that the tests in Theorem 4.2.1 lie in the complete class of tests. From (4.18) we have

$$\begin{aligned} f(\boldsymbol{N}) &= \beta(\boldsymbol{\psi}, \boldsymbol{c}, \boldsymbol{d}, \boldsymbol{e}, \boldsymbol{f}, \boldsymbol{g}) \exp(\Sigma_k^K \Sigma_i^{I-1} \Sigma_j^{J-1} \psi_{ij(k)} T_{ij(k)} \\ &+ \Sigma_i^{I-1} \Sigma_k^{K-1} n_{i+k} c_{ik} + \Sigma_j^{J-1} \Sigma_k^{K-1} n_{+jk} d_{jk} \\ &+ \Sigma_{i=1}^{I-1} n_{i++} e_i + \Sigma_{j=1}^{J-1} n_{+j+} f_j + \Sigma_{k=1}^{K-1} n_{++k} g_k) \cdot g(\boldsymbol{t}, \boldsymbol{m}). \end{aligned} \tag{4.34}$$

We rewrite (4.34) in the following family of distributions,

$$P(\boldsymbol{T}, \boldsymbol{Z}; \boldsymbol{\psi}, \boldsymbol{w}) = C(\boldsymbol{\psi}, \boldsymbol{w}) \exp[\boldsymbol{\psi}' \boldsymbol{T} + \boldsymbol{w}' \boldsymbol{Z}]. \tag{4.35}$$

That is, a random vector $(\boldsymbol{T}, \boldsymbol{Z}) \in \boldsymbol{R}^m \times \boldsymbol{R}^k$ has an exponential density. Let Θ denote the natural parameter space, and assume $(\mathbf{0}, \mathbf{0})$ is an interior point of Θ. Eaton described an essentially complete class of tests, and we need the following notation to formulate Eaton's result.

Let $\boldsymbol{\psi}$ be the parameter of interest and $\boldsymbol{w}$ be the nuisance parameters. The problem considered is that of testing hypotheses,

$$\begin{aligned} H_0 &: \boldsymbol{\psi} = \mathbf{0}, \\ H_a &: \boldsymbol{\psi} \in \Omega_1 \subseteq \boldsymbol{R}^m, \end{aligned}$$

where Ω_1 is contained in some half-space. It is assumed that for each $\boldsymbol{\psi} \in \Omega_1$, there exists a $\boldsymbol{w} \in \boldsymbol{R}^k$ such that $(\boldsymbol{\psi}, \boldsymbol{w}) \in \Theta$.

Let $V \subseteq \boldsymbol{R}^m$ be the smallest convex cone containing Ω_1, and let V^- denote the normal cone of V, e.g.,

$$V^- = \{\boldsymbol{u} \in \boldsymbol{R}^m : \Sigma_{j=1}^m u_i v_j \leq 0 \text{ for all } \boldsymbol{v} \in V\}, \qquad m = df. \tag{4.36}$$

Moreover, Φ stands for the class of nonempty closed convex sets in $\boldsymbol{R}^m$ and

$$\Phi(V) \quad = \quad \{C : C \in \Phi \text{ and } V^- \subset C - c \text{ for each } c \in \partial C\}, \tag{4.37}$$

where ∂C stands for the boundary of C.

Consider the set, $D^*(V)$, of test functions with the following property: if $\varphi \in D^*(V)$, there exits a measurable set $A \subseteq \boldsymbol{R}^m \times \boldsymbol{R}^k$ such that each $\boldsymbol{Z}$ section, $A(\boldsymbol{Z}) \subseteq \boldsymbol{R}^m$, is in $\Phi(V)$ and

$$\varphi(\boldsymbol{t}, \boldsymbol{z}) \quad = \quad \begin{cases} 1 & \text{if } \boldsymbol{T} \in A(\boldsymbol{Z})^C, \\ r(\boldsymbol{t}, \boldsymbol{z}) & \text{if } \boldsymbol{T} \in \partial A(\boldsymbol{Z}), \\ 0 & \text{if } \boldsymbol{T} \in \operatorname{Int} A(\boldsymbol{Z}), \end{cases}$$

where $A(\boldsymbol{Z})^C$ refers to the complement of $A(\boldsymbol{Z})$. The notation $A(\boldsymbol{Z})$ refers to $\boldsymbol{Z}$ section of acceptance region. This means the acceptance region at fixed $\boldsymbol{Z} = \boldsymbol{z}$ when we consider the conditional test. Eaton (1970) showed that D^* is an essentially complete class for testing $H_0 : \boldsymbol{\psi} = \boldsymbol{0}$ against $H_a : \boldsymbol{\psi} \in \Omega_1$. In light of (4.34), the testing problem in three-way contingency tables fits the framework of Eaton, which yields the fact that the tests in Theorem 4.2.1 lie in the complete class of tests.

4.5 Admissible Tests

Matthes and Truax (1967) described the class of admissible tests on multivariate exponential distributions for testing $H_0 : \boldsymbol{\psi} = \boldsymbol{0}$ against $H_a : \boldsymbol{\psi} \neq \boldsymbol{0}$, based on the conditional distribution of $\boldsymbol{T}$ given $\boldsymbol{Z}$. This description is given under the assumption that the support of conditional distribution is finite. They showed that a test φ is admissible if and only if there exists a convex acceptance region, say $A(\boldsymbol{Z})$, equivalent to $\varphi(\cdot, \boldsymbol{z})$, such that $\varphi(\cdot, \boldsymbol{z}) = 0$ at all nonextreme points of $A(\boldsymbol{Z})$. The notation $A(\boldsymbol{Z})$ refers to $\boldsymbol{Z}$ section of acceptance region, the same as in Section 4.4. Using methods

developed by Matthes and Truax, Ledwina (1978a, 1984) gave admissibility of tests on multivariate exponential distributions with discrete support. It is characterized by the fact that the conditional distribution of $\boldsymbol{T}$ given $\boldsymbol{Z} = \boldsymbol{z}$ is independent of the nuisance parameters $\boldsymbol{w}$. Hence, we consider the admissibility on each section of $\boldsymbol{Z} = \boldsymbol{z}$ separately, and then obtain the class of admissible tests for the original problem. The class of admissible tests for $H_0 : \boldsymbol{\psi} = \boldsymbol{0}$ against $H_a : \boldsymbol{\psi} \in \Omega_1 \subseteq \boldsymbol{R}^m$ in (4.35), based on the conditional distribution of $\boldsymbol{T}$ given $\boldsymbol{Z} = \boldsymbol{z}$, is described as follows. A test $\varphi(\boldsymbol{t})$ is admissible if and only if there exists a set $A \in \Phi(V)$ in (4.37) such that on each surface of $\boldsymbol{Z} = \boldsymbol{z}$, $A(\boldsymbol{Z}) \subseteq \boldsymbol{R}^m$ and

$$\varphi(\boldsymbol{t}) = \begin{cases} 1 & \text{if } \boldsymbol{T} \in A(\boldsymbol{Z})^C, \\ r(\boldsymbol{t}) & \text{if } \boldsymbol{T} \in E^{A(\boldsymbol{Z})}, \\ 0 & \text{if } \boldsymbol{T} \in \operatorname{Int} A(\boldsymbol{Z}), \end{cases}$$

where E^A denotes the set of all extreme points of A. This means that a test $\varphi(\boldsymbol{t})$ is admissible if and only if for each fixed $\boldsymbol{z}$, the acceptance region is convex, and randomization happens only at extreme points.

Ledwina (1984) also gave connections between admissibility of tests for the conditional distributions and the initial problem of tests based on (4.35). Ledwina showed that the test $\varphi(\boldsymbol{t}, \boldsymbol{z})$ is admissible for testing H_0 against H_a if and only if for every fixed $\boldsymbol{Z} = \boldsymbol{z}$, the test $\varphi(\cdot, \boldsymbol{z})$ is admissible in the class of tests based on the conditional distribution of $\boldsymbol{T}$ given $\boldsymbol{Z} = \boldsymbol{z}$. From the arguments in Ledwina, the tests in Theorem 4.2.1 are admissible. Hence, they are the exact, unbiased, and admissible tests in three-way tables.

4.6 Exact, Unbiased and Admissible Tests

In this section we illustrate the exact, unbiased, and admissible tests that satisfy the properties of Theorem 4.2.1. We discuss how to construct unbiased tests and how to set up critical regions to obtain tests of conditional independence of fixed size α, for the ordinal alternative. We focus on three-way tables where row and column classifications are ordinal, and the contents of Sections 3, 4 and 5 are combined together to give the unbiased and admissible tests. One advantage of ordinal models over the nominal-scale models is that tests based on ordinal models have more power to detect certain types of association and interaction (Agresti, 1990).

The model of homogeneous linear-by-linear association, which utilizes the ordinality of X and Y is

$$\log m_{ijk} = \mu + \lambda_i^X + \lambda_j^Y + \lambda_k^Z + \beta u_i v_j + \lambda_{ik}^{XZ} + \lambda_{jk}^{YZ}. \tag{4.38}$$

We test conditional independence, $H_0 : \psi_{ij(k)} = 0$, or equivalently, $\beta = 0$, against the alternative (4.38) of linear-by-linear association, using the sufficient statistic for β in that model,

$$T = \Sigma_k[\Sigma_i \Sigma_j u_i v_j n_{ijk}]. \tag{4.39}$$

We show that a test based on T satisfies the conditions of Theorem 4.2.1, so it is the exact, unbiased, and admissible test. First, we show that T can be expressed as $T = \Sigma_k[\sum_{i=1}^{I-1} \sum_{j=1}^{J-1} (u_i - u_{i+1})(v_j - v_{j+1}) t_{ij(k)}] + C$, where $t_{ij(k)} = \Sigma_{m=1}^{j} \Sigma_{l=1}^{i} n_{lmk}$, and C is a constant depending on the scores and the fixed marginal totals.

Let $u_{I+1} = v_{J+1} = 0$. Then,

$$\begin{aligned}
T &= \sum_k^K [\sum_i^I \sum_j^J u_i v_j n_{ijk}] \\
&= \sum_k^K [\sum_i^I \sum_j^J \{(u_i - u_{i+1}) + (u_{i+1} - u_{i+2}) + \cdots + (u_I - u_{I+1})\} \cdot \\
&\quad \{(v_j - v_{j+1}) + (v_{j+1} - v_{j+2}) + \cdots + (v_J - v_{J+1})\} n_{ijk}] \\
&= \sum_k^K [\sum_i^I \sum_j^J \{\sum_{a=i}^I (u_a - u_{a+1})\}\{\sum_{b=j}^J (v_b - v_{b+1})\} n_{ijk}] \\
&= \sum_k^K [\sum_i^I \sum_j^J \sum_{a=i}^I \sum_{b=j}^J (u_a - u_{a+1})(v_b - v_{b+1}) n_{ijk}] \\
&= \sum_k^K [\sum_{a=1}^I \sum_{b=1}^J \sum_{i=1}^a \sum_{j=1}^b (u_a - u_{a+1})(v_b - v_{b+1}) n_{ijk}] \\
&= \sum_k^K [\sum_{a=1}^I \sum_{b=1}^J (u_a - u_{a+1})(v_b - v_{b+1}) \sum_{i=1}^a \sum_{j=1}^b n_{ijk}] \\
&= \sum_k^K [\sum_{i=1}^I \sum_{j=1}^J (u_i - u_{i+1})(v_j - v_{j+1}) t_{ij(k)}] \\
&= \sum_k^K [\sum_{i=1}^{I-1} \sum_{j=1}^{J-1} (u_i - u_{i+1})(v_j - v_{j+1}) t_{ij(k)} \\
&+ (u_I - u_{I+1}) \sum_{j=1}^{J-1} (v_j - v_{j+1}) t_{Ij(k)} + (v_J - v_{J+1}) \sum_{i=1}^{I-1} (u_i - u_{i+1}) t_{iJ(k)} + u_I v_J t_{IJ(k)}] \\
&= \sum_k^K [\sum_{i=1}^{I-1} \sum_{j=1}^{J-1} (u_i - u_{i+1})(v_j - v_{j+1}) t_{ij(k)}] + C.
\end{aligned}$$

Thus, T is monotone in $\{t_{ij(k)}\}$ if the scores satisfy

$$(u_i - u_{i+1})(v_j - v_{j+1}) > 0, \tag{4.40}$$

for $i = 1, \cdots, I-1,\ j = 1, \cdots, J-1$; that is, if the scores $\{u_i\}, \{v_j\}$ are both monotone increasing or both monotone decreasing. We note that the statistic $\Sigma_k[\Sigma_{j=1}^{J}\Sigma_{i=1}^{I} t_{ij(k)}]$ is a special case of T, for the equally-spaced scores $\{u_i = I-(i-1)\}$, and $\{v_j = J-(j-1)\}$. Thus, tests based on T are unbiased.

In constructing critical regions, we utilize a secondary statistic, T', for ordering the tables for which $T = t_o$. The secondary statistic is used to generate a secondary partitioning to set up critical regions to obtain tests of conditional independence of fixed size α. When $I = J = 2$, we could use $T' = \sum X_k^2$ to order the tables for which $T = t_o$. The approach of Cohen and Sackrowitz (1992) is to utilize their conditional null probabilities to order the tables. These relate to the modified P-value, which we discussed in Chapter 2. The same argument applies if one uses some other secondary statistic. Let C_α be a constant, depending on $\boldsymbol{m}$, such that

$$P\{T \geq C_\alpha\} \geq \alpha \text{ and } P\{T > C_\alpha\} = \lambda < \alpha.$$

The test rejects if $T > C_\alpha$. When $T = C_\alpha$, consider all tables having $T = C_\alpha$, and order the tables according to their secondary test statistic values. When the large values of T' contradict the null, attention can be given to the tables having larger values of T' among the tables having $T = C_\alpha$. For another case, if some table has small probability under the null hypothesis, it implies that such a table would be unlikely to occur if H_0 is true. And for a particular value of T, a smaller table probability under the null corresponds to stronger contradiction to the null hypothesis. Hence, attention can be given to the tables whose null probabilities are less probable among them when we construct the rejection region using the null table probability for the secondary statistic. Thus, when $T = C_\alpha$, we reject for those tables whose secondary statistic values are the largest or whose probabilities are smallest, and whose probabilities total at most $(\alpha - \lambda)$. Instead of randomizing on all tables where $T = C_\alpha$, we allow randomization only at extreme points of a convex acceptance

section of the remaining points, so that the test is exact, unbiased, and admissible. We denote a test of this form by φ^*.

Forming the critical region in this way gives a test that is less likely to require randomization than the usual test φ that randomizes on the entire set $\{\mathbf{n} : \mathbf{m} \text{ fixed}, T = C_\alpha\}$. Also, the modified test is better than φ, since usually the entire set of tables having $T = C_\alpha$ contains nonextreme points, making φ inadmissible.

In this section, we have shown that a test based on T using monotone scores satisfies the properties of Theorem 4.2.1, since a test based on T has a desirable monotonicity property by the construction of T, and we allow randomization only at extreme points of the convex acceptance section. Hence, the test φ^* is exact, unbiased, and admissible. A nonrandomized test using T is unbiased and admissible, but it would be conservative when used with a fixed size α. But the test, φ^*, would have actual size closer to a nominal level than the ordinary test.

4.7 Example

We consider the test of conditional independence in three-way contingency tables, where row and column variables are ordinal. We assume that the model of no three-factor interaction holds and we can construct tests to increase power against important alternatives. We will illustrate construction of an exact, unbiased, and admissible test using $2 \times 2 \times 5$ contingency tables. When $I = J = 2$, the usual statistic $\Sigma_k n_{11k}$ results from the scores $u_1 = v_1 = 1$, $u_2 = v_2 = 0$ in $T = \Sigma_k[\Sigma_i \Sigma_j u_i v_j n_{ijk}]$. When $I = J = 2$, the test φ^* gives an alternative to the ordinary one for testing conditional independence for a set of 2×2 tables, under the assumption of a common odds ratio. The ordinary test is often inadmissible. For an $I \times J$ table, we can construct exact, unbiased, and admissible tests for an ordinal alternative to independence

by using a modified approach, but it is not easy to display the acceptance section if I and J are greater than 3.

4.7.1 Test of Conditional Independence : $2 \times 2 \times 5$ tables

We utilize the middle three subtables of Table 2.1 to illustrate construction of an exact, unbiased, and admissible test. We study size $\alpha = 0.05$ tests based on $T = \sum_k n_{11k}$. Given D and C marginal totals at each level of $P = \frac{1}{4}, \frac{1}{2}$ and 1, n_{112} can range between 0 and 3, n_{113} can range between 2 and 6, and n_{114} can have 5 or 6. The whole distribution of n_{112}, n_{113}, and n_{114} is composed of 40 tables. Since $P\{T \geq 13\} = 0.1136 > \alpha$ and $P\{T > 13\} = 0.0200 < \alpha$, randomization is required for those tables with $T = 13$. We use $\sum X_k^2$ or the null table probability for the secondary statistic. Followings are the tables with $T \geq 13$.

$$\begin{aligned}
1)T &= 15 \\
(n_{112}, n_{113}, n_{114}) &= (3,6,6) \text{ with } P(3,6,6) = \tfrac{2}{1452}, \textstyle\sum_k X_k^2 = 11.09 \\
2)T &= 14 \\
(n_{112}, n_{113}, n_{114}) &= \left\{ \begin{array}{ll} (2,6,6) & \text{with } P(2,6,6) = \frac{9}{1452}, \sum_k X_k^2 = 7.54 \\ (3,5,6) & \text{with } P(3,5,6) = \frac{16}{1452}, \sum_k X_k^2 = 6.59 \\ (3,6,5) & \text{with } P(3,6,5) = \frac{2}{1452}, \sum_k X_k^2 = 11.09 \end{array} \right. \\
3)T &= 13 \\
(n_{112}, n_{113}, n_{114}) &= \left\{ \begin{array}{ll} (1,6,6) & \text{with } P(1,6,6) = \frac{9}{1452}, \sum_k X_k^2 = 7.54 \\ (2,6,5) & \text{with } P(2,6,5) = \frac{9}{1452}, \sum_k X_k^2 = 7.54 \\ (3,5,5) & \text{with } P(3,5,5) = \frac{16}{1452}, \sum_k X_k^2 = 6.59 \\ (3,4,6) & \text{with } P(3,4,6) = \frac{30}{1452}, \sum_k X_k^2 = 5.09 \\ (2,5,6) & \text{with } P(2,5,6) = \frac{72}{1452}, \sum_k X_k^2 = 3.04. \end{array} \right.
\end{aligned}$$

The usual 0.05-size conditional test based on T is

$$\varphi = \begin{cases} 1 & \text{if } (n_{112}, n_{113}, n_{114}) = (3,6,6), (2,6,6), (3,5,6), (3,6,5) \\ 0.3206 & \text{if } (n_{112}, n_{113}, n_{114}) = (1,6,6), (2,5,6), (3,4,6), (2,6,5), (3,5,5) \\ 0 & \text{otherwise.} \end{cases}$$

This test randomizes with equal probability on all tables for which $T = 13$. Since the table (2,5,6) is an interior point of line segment between tables (1,6,6) and (3,4,6), it is not an extreme point of a convex acceptance region. It makes φ inadmissible by noting that randomization should occur only at extreme points in order to be admissible. Hence, another test φ' will beat the test φ.

$$\varphi' = \begin{cases} 1 & \text{if } (n_{112}, n_{113}, n_{114}) = (3,6,6), (2,6,6), (3,5,6), (3,6,5), (1,6,6) \\ 0.2724 & \text{if } (n_{112}, n_{113}, n_{114}) = (2,5,6), (3,4,6), (2,6,5), (3,5,5) \\ 0 & \text{otherwise.} \end{cases}$$

Since the table (1,6,6) has the largest $\sum_k X_k^2$ value or the smallest null table probability among tables for which $T = 13$, it can be included in the rejection region. The table (2,5,6) is now an extreme point for this test. Since randomization is permitted only on the extreme points of convex acceptance region, it is admissible. The exact test φ^* that orders the tables according to their secondary statistic values is

$$\varphi^* = \begin{cases} 1 & \text{if } (n_{112}, n_{113}, n_{114}) = (3,6,6), (2,6,6), (3,5,6), (3,6,5), (1,6,6), \\ & \qquad\qquad\qquad\qquad (2,6,5), (3,5,5) \\ 0.3200 & \text{if } (n_{112}, n_{113}, n_{114}) = (3,4,6) \\ 0 & \text{otherwise.} \end{cases}$$

We can add tables into the rejection region until the probability of rejection is not greater than the size. Hence, two tables (2,6,5) and (3,5,5) are entered into the rejection region since they have the next largest $\sum_k X^2$ values or the next smallest null table probabilities. Furthermore, the table (2,5,6), which has the table probability

close to our size, can be excluded from randomization so that the table (3,4,6) is the only extreme point for possible randomization. The test φ^* randomizes only on an extreme point (3,4,6) of its convex acceptance region, and it satisfies the properties of Theorem 4.2.1. Hence, it is exact, unbiased, and admissible. Compared to the previous test, it has the advantage of having only a single table for which randomization is necessary. The probability that randomization is required is only 0.0207, rather than 0.0937. In this data set, we get the same results of exact, unbiased, and admissible tests using either $\sum_k X_k^2$ or the null table probability for the secondary statistic.

4.8 Discussion

For $I \times J \times K$ tables, we generalized results of Cohen and Sackrowitz (1992) and showed how to construct exact, unbiased, and admissible tests for an ordinal alternative to conditional independence. The ordinary exact test of conditional independence for $2 \times 2 \times K$ tables is often inadmissible. In practice, randomized tests are unacceptable. Thus, even the tests described in Section 6 that require less randomization than usual are not intended for practical use. However, results of that section suggest an obvious way of forming critical regions for tests so that one can have actual size closer to a desired value (such as 0.05) than would be possible with the ordinary test.

CHAPTER 5
CONCLUSION

5.1 Discussion

The conservativeness due to the discreteness of a statistic is a typical problem for exact inference with categorical data. Ways of reducing the conservativeness in exact tests and confidence intervals were proposed in Chapter 2. We prefer modified exact tests and confidence intervals to the ordinary exact ones because they are less conservative than the ordinary ones, but still guarantee at least the nominal level. We also prefer confidence intervals based on inverting two-sided tests over those based on inverting two separate one-sided tests because they tend to be less conservative.

The approach using a modified P-value can be utilized in approximating exact inference regarding conditional associations in $I \times J \times K$ tables. In Chapter 3 we discussed six test statistics for conditional independence. We obtained precise estimates of ordinary and modified exact P-values by using a simulation algorithm for cases that currently are computationally infeasible.

For $I \times J \times K$ tables, Chapter 4 discussed construction of tests of conditional independence that are exact, unbiased, and admissible for an ordinal alternative. By using a modified approach, less randomization is required than usual, and we obtain actual size closer to a nominal level. The ordinary exact test of conditional independence for $2 \times 2 \times K$ tables is often inadmissible, and we showed how to obtain improved tests.

5.2 Future Research

We have considered improved "exact" inference about conditional association in $2 \times 2 \times K$ contingency tables. The idea of a modified P-value can be applied to any contingency tables, and it can be calculated for any test statistic having a discrete distribution. One research study could be the application of the modified approach to exact tests of no three-factor interaction. Zelen (1971) presented an exact test of homogeneity of odds ratios in $2 \times 2 \times K$ tables. For an exact test of no three-factor interaction for $2 \times 2 \times K$ tables, an efficient score statistic against the saturated model is the Pearson statistic for testing the fit of that model (Agresti 1992). We could use this score statistic as a primary statistic and the table probability as a secondary statistic to define modified P-values. We could study how much improvement can be obtained by using a modified approach.

We could consider a modified confidence interval for the β parameter in the linear-by-linear association model. Under the alternative, the conditional distribution of $T = \sum\sum u_i v_j n_{ij}$ has a noncentral hypergeometric distribution (2.10), where $e^{\beta} = \theta$, and c_t is the sum of $(\Pi\Pi n_{ij}!)^{-1}$ for all tables with given marginal distributions having $T = t_o$ (Agresti *et al.* 1990). By using a modified confidence interval, we could reduce the conservativeness of the Agresti-Mehta-Patel interval.

As we mentioned in Section 2.4.1 for $2 \times 2 \times K$ tables, we could base confidence intervals on tests in which the two-sided P-value uses a non-null test statistic, instead of the table probability. For instance, we could consider a test statistic

$$T(\theta) = \frac{|\sum n_{11k} - \sum m_{11k}|^2}{\sum V(n_{11k})},$$

where under the alternative of assuming θ, m_{11k} is the mean of n_{11k}, and $V(n_{11k})$ is its variance. Since for a fixed value of θ, $\sum V(n_{11k})$ is a constant, $T(\theta)$ depends

only on its numerator. By using the exact non-null distribution, we could construct a two-sided ordinary or modified confidence interval.

Another area to consider is how we can apply important sampling (Mehta, Patel, and Senchaudhuri 1988) as an alternative to conventional Monte Carlo sampling to simulate the exact distribution and to estimate exact significance levels. In importance sampling, the tables are selected in proportion to their importance for reducing the variance of the estimated Monte Carlo P-values, whereas in Monte Carlo sampling, the tables are sampled independently with replacement from the reference set. The accuracy and the speed will be increased by using importance sampling.

We could use a simulation algorithm to approximate exact confidence intervals. Then, we need to have an algorithm to simulate the non-null distribution. Under the alternative, the joint probability distribution of a table has a noncentral hypergeometric distribution, and random tables should satisfy the association structure as well as the fixed margins. As we construct an "exact" confidence interval for a parameter by inverting the results of the exact conditional tests based on ordinary or modified exact P-values, we can approximate exact confidence intervals for a parameter by the same method based on the estimate of ordinary or modified exact P-values.

Also, we could approximate exact inference for the test of no three-factor interaction. In this case the conditional reference set is the set of $I \times J \times K$ tables whose XY, XZ, YZ marginal tables are fixed at the corresponding values of the observed tables. More power would be obtained for narrower alternatives that utilize ordinality.

For the test of conditional independence in $I \times J \times K$ tables, we defined the class of exact, unbiased, and admissible tests. There are other null hypotheses of interest. We could consider the class of exact, unbiased, and admissible tests for testing no three-factor interaction against an ordinal alternative.

In summary, we suggested exact inference regarding conditional associations in three-way tables, modifying the usual exact conditional approach. This seems to be a

promising approach for categorical data analysis, and more work can be done utilizing this approach.

APPENDIX A
SOURCE CODE FOR EXACT INFERENCE

Following are FORTRAN source code for computing the ordinary and modified exact P-values, four types of confidence intervals, and coverage probability. Data or its file name can be entered from console, and this program provides four types of confidence intervals or coverage probability by the option. When the coverage probability is requested, it makes five output files. They are "OO.CI" for one-sided ordinary exact confidence interval, "OM.CI" for one-sided modified exact confidence interval, "TO.CI" for two-sided ordinary exact confidence interval, "TM.CI" for two-sided modified exact confidence interval, and "COVER.P" for coverage probability for four types of confidence intervals. This program, for $2 \times 2 \times K$ tables, is an adaptation of one written by Vollset and Hirji (1991) for ordinary exact inference.

```
      integer itab(1000,4),IOTOT
      INTEGER NIK(2,1000),NJK(2,1000),NTOT(1000)
c======================================================================
      INTEGER ISUMA,J,SCD
      INTEGER*2 JH3,JM3,JS3,JSS3
      integer infhyl(1000),infhyu(1000),INUM(270000,20),INUM1(270000,1)

c======================================================================
      double precision hyp(0:2000),ds(0:1,0:5500),dd1,lge

c======================================================================
      DOUBLE PRECISION C(5500),B,LLL,K,FF,R1(5),R2(5)
      DOUBLE PRECISION ROOTRF,EPS,X0,X1,X3
      DOUBLE PRECISION LL,UL,MH,MUE,KA(100,2),RUL,RLL,MIDP,MAXP,PVAL2
      DOUBLE PRECISION FLOWER,FUPPER,FLOWBO,FUPPBO,MAXPE,pobsh,ODR
      DOUBLE PRECISION ALPHA,VRBG,SVRBG,ALL,AUL,START,ELL,EUL,ELL1,EUL1
      DOUBLE PRECISION PALPHA,P_UP,P_LO,P_UP1,P_LO1
```

```
      DOUBLE PRECISION P_UP2,P_LO2,ELL2,EUL2
      DOUBLE PRECISION OOCI(270000,2),OMCI(270000,2),TOCI(270000,2)
      DOUBLE PRECISION TMCI(270000,2),COVER(1600,5)
      DOUBLE PRECISION HYPD10(270000,1),POOCI,POMCI,PTOCI,PTMCI
      DOUBLE PRECISION ELL3,EUL3,P_LO3,P_UP3,DALL
      DOUBLE PRECISION DENO,hypd(1000,0:2000),POBSH1,PEXIMP,PEX
      DOUBLE PRECISION HYPD2(270000,20),HYPD1(270000,1)
      DOUBLE PRECISION CHI(270000),CHIOBS

      CHARACTER*16 FNAME
      COMMON/PARAM/C,J,SCD,K,FF
      COMMON/CI1/ik,mxs,mxz,mxd,lge,itab,hyp,ds,ipar,k1,k2,ierr,pobsh
      COMMON /CH/ NIK,NJK,NTOT
      COMMON /ART/ IOTOT
      COMMON/CI2/hypd,infhyl,infhyu,POBSH1,PEXIMP,PEX
      COMMON /DKIM/ DENO,ITOT,ISUML,INUM,HYPD2,INUM1,HYPD1
      COMMON /CH1/ CHI,CHIOBS

      EXTERNAL FLOWER,FUPPER,FLOWBO,FUPPBO
c======================================================================
C
c      DATA LGE /307.0D+00/
c======================================================================
      DATA KA(95,1) /3.84145882D+00 / KA(95,2) /2.5D-02/
      DATA KA(90,1) /2.70554345D+00 / KA(90,2) /5.0D-02/
      DATA KA(99,1) /6.63489660D+00 / KA(99,2) /5.0D-03/
      DATA KA(80,1) /1.64237442D+00 / KA(80,2) /1.0D-01/
      DATA KA(50,1) /0.45493642D+00 / KA(50,2) /2.5D-01/
      DATA MIDP /0.D00/, MAXP /0.D00/
C
C     FF IS 1 FOR EXACT AND 0.5 FOR MID-P EXACT
C     IMAX MAX NO. OF ITERATIONS
C     K ALPHA/2
C     EPS STOPPING CRITERION
C

      LGE=307.0D+00

      WRITE(*,10000)
10000 FORMAT(3(/),T12,'***** Ex2x2xK (version 24.0 -- 5/94) *****',/,
     1       /,T12,'Ordinary and Modified Exact P-values and CIs',/,
     2          T12,'for several 2x2 tables.',/,
     3          T12,'  One-sided and Two-sided Approach : ',/)

      write(*,10001)
```

```
10001 FORMAT(T7,'    This program calculates',/,
     1       T7,' 1. Ordinary and Modified Exact P-values, ',/,
     2       T7,' 2. Four Types of Exact Confidence Limits',/,
     3       T7,'    for the Common Odds Ratio, and',/,
     4       T7,' 3. Coverage Probability for CIs.')

      WRITE(*,10002)
10002 FORMAT(/,T7,'    The program of Vollset, Hirji, Elashoff is ',
     +  /,T7,' graciously provided and slightly modified.',
     +  /,T7,' Several routines are added for modified ',
     +        'exact inference.')

      WRITE(*,10004)
10004 FORMAT( /,/,T7,' Any questions about the use of this software ',
     +  /,T7,' can be directed to Dr. Alan Agresti or Donguk Kim.')


C--------------------1-ALPHA/2-----
      IA = 95
C---------------------------------
      ALPHA=1.D0-DBLE(IA)/100.D0

1     FF = 0.D+00
      IK=0
      IMAX=50
      K = KA(IA,2)
      EPS = 1.D-09
c==================================================================

c
c     intrinsic functions : float();dexp()
c
      iin=5
      iot=6
c
c     maximum number of strata = 1000
c     maximum value of range of
c     hypergeometric distribution = 2000
c     maximum value of range of
c     final distribution = 5500
c
      mxs = 1000
      mxz = 2000
      mxd = 5500
```

```
c
c     maximum stratum size
c
      mxss = 500000



c======================================================================
C
C     READ DATA
C

10    WRITE(IOT,999)
      WRITE(IOT,997)
      READ(IIN,15)FNAME
C--------------------------------------------
C      FNAME='peni.dat'
C--------------------------------------------

15    FORMAT(A16)
      IF(FNAME .EQ. 'c' .OR. FNAME .EQ. 'C')THEN
          PRINT *,'GIVE 1-ALPHA/2: 50,80,90,95 OR 99)'
          READ(IIN,16)IA
16        FORMAT(I2)
          GOTO 1
      ENDIF
      IF(FNAME .EQ. 'k' .OR. FNAME .EQ. 'K')THEN
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
18        write(iot,20)
20        format(/10x,'Enter no. of strata')
          read(iin,*)ik
          if (ik .lt. 1) goto 10
          open(unit=28,file='2x2.dat')
          do 30 i=1,ik
             write(iot,40)i
40           format(/10x,'Enter table',1x,i3)
             read(iin,*)itab(i,1),itab(i,2),itab(i,3),itab(i,4)
             write(iot,*)(itab(i,j),j=1,4)
             write(28,*)(itab(i,j),j=1,4)

      NIK(1,I)=ITAB(I,1)+ITAB(I,2)
      NIK(2,I)=ITAB(I,3)+ITAB(I,4)
      NJK(1,I)=ITAB(I,1)+ITAB(I,3)
      NJK(2,I)=ITAB(I,2)+ITAB(I,4)
```

```
      NTOT(I)=NIK(1,I)+NIK(2,I)

 30        continue
           GOTO 100
      ENDIF
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      OPEN(UNIT=27,FILE=FNAME)

C-----------------------------------------------
C      OPEN(UNIT=27,FILE='peni.dat')
C-----------------------------------------------

      DO 70 I=1,MXS
            READ(27,*,END=100)(ITAB(I,J),J=1,4)
            IK=IK+1
            WRITE(IOT,*)(ITAB(I,J),J=1,4)

      NIK(1,I)=ITAB(I,1)+ITAB(I,2)
      NIK(2,I)=ITAB(I,3)+ITAB(I,4)
      NJK(1,I)=ITAB(I,1)+ITAB(I,3)
      NJK(2,I)=ITAB(I,2)+ITAB(I,4)
      NTOT(I)=NIK(1,I)+NIK(2,I)

70    CONTINUE

100   PRINT *,'NO.STRATA',IK
      WRITE(*,80)
80    FORMAT(/,'ENTER CODE FOR ANALYSIS :',
     1    /,/,'       1: P-VALUE AND CONFIDENCE INETRVAL',
     2       /,'        2: COVERAGE PROBABILITY FOR CIS.',/)
      READ(*,*)NCODE
      IF (NCODE .EQ. 1) GO TO 110

      IERR = 0
      jci=0
      call cnv2x2(ik,mxs,mxz,mxd,lge,itab,hyp,ds,ipar,k1,k2,ierr,pobsh,
     1 jci,odr)

       print*
       print*,'TOTAL NO. OF RANDOM TABLES =',iotot

C*******************************************************************
C
C  COMPUTE CIS FOR EACH RANDOM TABLE
C
```

```
C*****************************************************************
        OPEN(UNIT=45,FILE='OO.CI')
        OPEN(UNIT=46,FILE='OM.CI')
        OPEN(UNIT=47,FILE='TO.CI')
        OPEN(UNIT=48,FILE='TM.CI')

      DO 5000 IAITN=1,IOTOT
C       print*,'data'
C       PRINT*,'NO. OF RANDOM TABLE =',IAITN

      DO 5010 I=1,IK
      ITAB(I,1)=INUM(IAITN,I)+INFHYL(I)
      ITAB(I,2)=NIK(1,I)-ITAB(I,1)
      ITAB(I,3)=NJK(1,I)-ITAB(I,1)
      ITAB(I,4)=NTOT(I)-(ITAB(I,1)+ITAB(I,2)+ITAB(I,3))
C       print*,itab(i,1),itab(i,2),itab(i,3),itab(i,4)
5010  CONTINUE

      ff=0.d0

110   IERR = 0
C      CALL GETTIM(JH1,JM1,JS1,JSS1)
      jci=0
      call cnv2x2(ik,mxs,mxz,mxd,lge,itab,hyp,ds,ipar,k1,k2,ierr,pobsh,
     1 jci,odr)

      IF (IERR .GT. 0) THEN
         CALL ERROR(IERR,MXs,mxss,MXZ,MXD)
         GOTO 170
      ENDIF

C      CALL GETTIM(JH2,JM2,JS2,JSS2)
C      ITIME = 60*60*(JH2-JH1) + 60*(JM2-JM1) + JS2-JS1
c      ATIME = FLOAT(ITIME) + FLOAT(JSS2-JSS1)/100.0
170   CONTINUE

C=========================================================
C  CALCULATES OBSERVED POSITION IN SAMPLE SPACE
C  J   - POSITION
C  SCD - SIZE COND. SAMPLE SPACE

C  ADDED BY DONGUK KIM
      ISUMA=0
C
      DO 180 I=1,IK
```

```
          ISUMA = ISUMA + ITAB(I,1)
 180    CONTINUE

        J = ISUMA - K1 + 1
        SCD = K2 - K1 + 1

        IF (NCODE .EQ. 1) THEN
          print*
          PRINT*,'-----------------------------------------------'
          print*,'DISTRIBUTION in T : ',k1,' <= T <=',k2,scd,' values'
          print*,'OBSERVED T is in ',j,' th position among ',scd
          print*,'OBSERVED PRIMARY TEST STATISTIC =',isuma
          print*,'OBSERVED SECONDARY TEST STATISTIC =',SNGL(CHIOBS)
          PRINT*,'-----------------------------------------------'
          print*
          PRINT*,'-----------------------------------------------'
          WRITE(*,906)PEXIMP
          WRITE(*,907)PEX
          PRINT*
          PRINT*,'PROB. OF OBSERVED TABLES =',SNGL(POBSH1/DENO)
          PRINT*,'-----------------------------------------------'
          PRINT*
 906    FORMAT('THE MODIFIED EXACT P-VALUE =',3X,F12.6)
 907    FORMAT('THE ORDINARY EXACT P-VALUE =',3X,F12.6)
          ENDIF

          DO 210 I=K1,K2
          C(I-K1+1)=DS(IPAR,I-k1)
 210      CONTINUE

        IF (NCODE .EQ. 1) THEN
          open(unit=29,file='dist.fx5')
          DO 211 I=K1,K2
        write(29,*)i-k1+1,C(i-k1+1),DEXP(C(i-k1+1)),I
 211      CONTINUE
          ENDIF

C
C       IPOS=1 IF OBSERVED IS ON LOWER BOUNDARY, 2 ON UPPER, 0 OW
C
        IPOS=0
        IF(J .EQ. 1) IPOS =1
        IF(J .EQ. SCD) IPOS=2
C        PRINT *,'IPOS',IPOS
C
```

```
C CALCULATE STARTING VALUES
C
C       CALL SATO(ITAB,IK,LL,UL,MH,KA,IA,RLL,RUL,IPOS)
      CALL SATO(ITAB,IK,LL,UL,MH,KA,IA,RLL,RUL,IPOS,VRBG)

      R1(3) = UL
      R2(3) = LL
      R1(4) = RUL
      R2(4) = RLL
C-----P-VALUES----------------------
      IF(IPOS .GE.1)GOTO 220
      FF=0.5
      MIDP=FLOWER(0.D00)+K
      PVAL2=FUPPER(0.D00)+K
      IF(MIDP .GT. PVAL2)MIDP=PVAL2
      MIDP=MIDP*2
      FF=1.0
      MAXP=FLOWER(0.D00)+K
      PVAL2=FUPPER(0.D00)+K
      IF(MAXP .GT. PVAL2)MAXP=PVAL2
      MAXPE=MAXP
C      PRINT*,'ONE SIDED P_EXACT ',MAXPE
C
      MAXP=MAXP*2
220   FF=1.0
      IF(IPOS .EQ. 1)THEN
           MAXP=2*(FLOWBO(0.D00)+K)
           FF=0.5
           MIDP=2*(FLOWBO(0.D00)+K)
      ENDIF
      FF=0.5
      IF(IPOS .EQ. 2)THEN
           MIDP=2*(FUPPBO(0.D00)+K)
           FF=1.0
           MAXP=2*(FUPPBO(0.D00) +K)
      ENDIF
      FF=0
C-----------------------------------
      DO 1000 JJ=1,2
          FF=FF+0.5
C          PRINT *,'FF',FF
          IF(IPOS .EQ. 1) GOTO 300
          IF(IPOS .EQ. 2) GOTO 310
C---------M.U.E
          IF(FF .EQ. 0.5) THEN
```

```
                 K  = 4.999999999999999D-001
                 X0 = DLOG(MH)
C                 PRINT *,'MH',MH
                 CALL brent(X0,EPS,IMAX,ROOTRF,FLOWER,NRF)
                 MUE = ROOTRF
C                 PRINT *,'M.U.E',ROOTRF
                 K = KA(IA,2)
C                 PRINT *,'K',K
           END IF
C-----------------------------------
           X0 = DLOG(UL)
           CALL brent(X0,EPS,IMAX,ROOTRF,FLOWER,NRF)
           R1(JJ)=ROOTRF
           X0 = DLOG(LL)
           CALL brent(X0,EPS,IMAX,ROOTRF,FUPPER,NRF)
           R2(JJ)=ROOTRF
           GOTO 340
 300       X0 = DLOG(UL)
           CALL brent(X0,EPS,IMAX,ROOTRF,FLOWBO,NRF)
           R1(JJ)=ROOTRF
           GOTO 341
 310       X0 = DLOG(LL)
           CALL brent(X0,EPS,IMAX,ROOTRF,FUPPBO,NRF)
           R1(JJ)=ROOTRF
           GOTO 342
C---------------------------
 340       IF(FF .EQ. 0.5) GOTO 1000

         IF (NCODE .EQ.1) THEN
           WRITE(*,999)IA
           PRINT *,' POINT ESTIMATES'
           PRINT *,' '
           WRITE(*,979)MH,DEXP(MUE)
           PRINT *,' '
           WRITE(*,993)MAXPE
           WRITE(*,994)POBSH
           PRINT *,' '

           PRINT *,' INTERVAL ESTIMATES           LOWER          UPPER
    +         2*ONESIDED P'
           PRINT *,' '
           WRITE(*,991)DEXP(R2(2)),DEXP(R1(2)),MAXP
           WRITE(*,992)DEXP(R2(1)),DEXP(R1(1)),MIDP
           WRITE(*,980)R2(3),R1(3)
           WRITE(*,981)R2(4),R1(4)
```

```
      ENDIF

        GOTO 343
 341    IF (FF .EQ. 0.5)GOTO 1000

      IF (NCODE .EQ.1) THEN
        WRITE(*,999)IA
        PRINT *,' LOWER BOUNDARY: UPPER LIMITS ONLY'
        PRINT *,' '
        PRINT *,' INTERVAL ESTIMATES              LOWER          UPPER
     +     2*ONESIDED P'
        WRITE(*,971)DEXP(R1(2)),MAXP
        WRITE(*,972)DEXP(R1(1)),MIDP
        WRITE(*,973)R1(3)
      ENDIF

        GOTO 343
 342    IF(FF .EQ. 0.5)GOTO 1000

      IF (NCODE .EQ. 1) THEN
        WRITE(*,999)IA
        PRINT *,' UPPER BOUNDARY: LOWER LIMITS OMLY'
        PRINT *,' '
        PRINT *,' INTERVAL ESTIMATES              LOWER          UPPER
     +     2*ONESIDED P'
        WRITE(*,974)DEXP(R1(2)),MAXP
        WRITE(*,975)DEXP(R1(1)),MIDP
        WRITE(*,976)R2(3)
      ENDIF

 343     CONTINUE
      IF (NCODE .EQ. 1) THEN
        PRINT*,'---------------------------------------------------'
      ENDIF

 971    FORMAT('  MAX-P EXACT           ',17X,2(5X,F12.6))
 972    FORMAT('  MID-P EXACT           ',17X,2(5X,F12.6))
 973    FORMAT('  MANTEL-HAENSZEL-SATO',17X,5X,F12.6)
 974    FORMAT('  MAX-P EXACT           ',5X,F12.6,17X,F12.6)
 975    FORMAT('  MID-P EXACT           ',5X,F12.6,17X,F12.6)
 976    FORMAT('  MANTEL-HAENSZEL-SATO',5X,F12.6)
 979    FORMAT('  MANTEL-HAENSZEL = ',F12.6,
     +    '   MEDIAN UNBIASED =',F12.6)
 980    FORMAT('  MANTEL-HAENSZEL-SATO',2(5X,F12.6))
 981    FORMAT('  MANTEL-HAENSZEL-RBG ',2(5X,F12.6))
```

```
 988       FORMAT(/,/,'  MODIFED P-VALUE ',3(5X,F12.6))
 990       FORMAT('MODIFED EXACT CI USING BRENT1',2(5X,F12.6))
C 990       FORMAT(/,/,'  MID P CORRECTED P EXACT      ',3(5X,F12.6))
 991       FORMAT('  MAX-P EXACT          ',3(5X,F12.6))
 992       FORMAT('  MID-P EXACT          ',3(5X,F12.6))
 993       FORMAT(' ONE SIDED P EXACT   ',5X,F12.6)
 994       FORMAT(' PROB OF OBSERVED TABLES',2X,F12.6)
 995       FORMAT(2(10X,F12.6))
 996       FORMAT(10X,F12.6)
 997       FORMAT(10X,'ENTER  FILENAME (k for keyboard entry - c to chan
     +ge alpha-level)',4(/))

 998       FORMAT(/2X,'ELAPSED TIME (SECS) = ',F8.2,' + ',F8.2,' = ',
     +     F8.2)
 999       FORMAT(3(/),2X,I2,
     +     '% TWO-SIDED EXACT CONFIDENCE',
     +     ' LIMITS FOR THE COMMON ODDS RATIO',
     +     2(/))
1000  CONTINUE

       IF (NCODE .EQ. 1) THEN
         WRITE(*,350)
350      FORMAT(/,/,'MODIFIED EXACT CONFIDENCE INTERVAL (Y=1,N=0) ?',/)
         READ(*,*)NCODE1
           IF (NCODE1 .NE. 1) THEN
              PRINT*,'END'
              GO TO 1002
           ENDIF
      ENDIF

      JCI=1

C****************************************************************
C  ONE-SIDED MODIFIED P CONFIDENCE INTERVAL.
C****************************************************************

2001  CONTINUE
C2001  WRITE(*,2005)
2005  FORMAT(/,/,'MODIFIED EXACT CONFIDENCE LIMITS FOR ',/,
     1 'THE COMMON ODDS RATIO USING ITERA (Y=1,N=0) ?',/,/)
c      READ(*,*)JSCI1
       JSCI1=1
C       PRINT*,'JSCI1=1'

      IF (JSCI1 .EQ. 0) GO TO 3001
```

```
C  INITIAL VALUE FOR ITERA1 IS THE LIMITS FROM ORDINARY EXACT CI.
        ALL=DEXP(R2(2))
        AUL=DEXP(R1(2))*1.1D0
C        AUL=DEXP(R1(2))
C        print*,'INITIAL VALUE USING ORDINARY EXACT CI = ',all,aul

C  COMPUTE LOWER LIMIT
        ist=1
        JCIO=2

        IF (J .EQ. SCD) ALL=DEXP(R1(2))

        IF (J .EQ. 1) THEN
          ELL1=0.D0
          P_LO1=1.D0
C          PRINT*,'LOWER LIMIT =',ELL1
C          PRINT*
          GO TO 2006
        ENDIF

C       PRINT*,'INITIAL VALUE FOR THE LOWER LIMIT = ',ALL
        CALL ITERA1(ALPHA,ALL,ELL1,ist,JCIO,PALPHA)
        P_LO1=PALPHA
C        print*,'lower limit ell1 from ITERA1 =',ell1
C        PRINT*

C  COMPUTE UPPER LIMIT
c        START=1.D0/AUL
2006    START=AUL
        ist=2
        JCIO=1

        IF (J .EQ. SCD) THEN
          EUL1=99999.999999
          P_UP1=1.D0
C          PRINT*,'UPPER LIMIT =',EUL1
          GO TO 2007
        ENDIF

C       PRINT*,'INITIAL VALUE FOR THE UPPER LIMIT = ',AUL
C        print*,'start=',start
        CALL ITERA1(ALPHA,START,EUL1,ist,JCIO,PALPHA)
        P_UP1=PALPHA
C        print*,'upper limit eul1 from ITERA1=',eul1
2007    CONTINUE
```

```
       IF (NCODE .EQ. 2) GO TO 3001
       PRINT*
      PRINT*,'-------------------------------------------------'
      WRITE(*,2010) ELL1,EUL1
      PRINT*,'P-VALUE FOR THE LIMIT (low,up) =',P_LO1,P_UP1
      PRINT*,'-------------------------------------------------'
      PRINT*
 2010 FORMAT('ONE-SIDED MODIFIED EXACT CI  ',2(5X,F12.6))


C**************************************************************
C     TWO-SIDED ORDINARY P CONFIDENCE INTERVAL.
C
C  The P-value is the sum of the either tail.
C**************************************************************

3001  CONTINUE
C3001  WRITE(*,3600)
3600   FORMAT(/,/,'TWO-SIDED ORDINARY EXACT CONFIDENCE LIMITS ',
     1 'FOR THE COMMON ODDS RATIO (Y=1,N=0) ?',/)
c      READ(*,*)JTSCI
       IOOTO=1
       JTSCI=1
C       PRINT*,'JTSCI=1'

      IF (JTSCI .EQ. 0) GO TO 1001

C  STARTING VALUES ARE LIMITS FOR ORDINARY EXACT CI.
       ALL=DEXP(R2(2))
       AUL=DEXP(R1(2))*1.1D0
C       AUL=DEXP(R1(2))

C       print*,'INITIAL VALUE USING ORDINARY EXACT CI = ',all,aul

C  COMPUTE LOWER LIMIT
      ist=1

      IF (J .EQ. 1) THEN
        ELL3=0.D0
        P_LO3=1.D0
C         PRINT*,'LOWER LIMIT =',ELL3
C         PRINT*
        GO TO 3006
      ENDIF
```

```
      IF (J .EQ. SCD) ALL=DEXP(R1(2))

C       PRINT*,'INITIAL VALUE FOR THE LOWER LIMIT = ',ALL
        CALL ITERA(ALPHA,ALL,ELL3,ist,PALPHA,IOOTO)
        P_LO3=PALPHA
C        print*,'lower limit ell3 from ITERA =',ell3
C        PRINT*

C  COMPUTE UPPER LIMIT
c        START=1.D0/AUL
3006   START=AUL
        ist=2

       IF (J .EQ. SCD) THEN
         EUL3=99999.999999
         P_UP3=1.D0
C         PRINT*,'UPPER LIMIT =',EUL3
         GO TO 3007
       ENDIF

C       print*,'start=',start
C      PRINT*,'INITIAL VALUE FOR THE UPPER LIMIT = ',AUL
       CALL ITERA(ALPHA,START,EUL3,ist,PALPHA,IOOTO)
       P_UP3=PALPHA
C      print*,'upper limit eul3 from ITERA =',eul3

3007  CONTINUE

        IF (NCODE .EQ. 2) GO TO 1001
      PRINT*
      PRINT*,'-----------------------------------------------'
      WRITE(*,3989) ELL3,EUL3
      PRINT*,'P-VALUE FOR THE LIMIT (low,up) =',P_LO3,P_UP3
      PRINT*,'-----------------------------------------------'
      PRINT*
3989      FORMAT('TWO-SIDED ORDINARY EXACT CI  ',2(5X,F12.6))


C*****************************************************************
C     TWO-SIDED MODIFIED P CONFIDENCE INTERVAL.
C
C  The P-value is the sum of the either tail.
C*****************************************************************
```

```
1001  CONTINUE
C1001  WRITE(*,600)
600   FORMAT(/,/,'TWO-SIDED MODIFIED EXACT CONFIDENCE LIMITS ',
     1 'FOR THE COMMON ODDS RATIO (Y=1,N=0) ?',/)
c       READ(*,*)JSCI
        IOOTO=2
        JSCI=1
C        PRINT*,'JSCI=1'

      IF (JSCI .EQ. 0) GO TO 1002

C  STARTING VALUES ARE LIMITS FOR ORDINARY EXACT CI.
        ALL=DEXP(R2(2))
        AUL=DEXP(R1(2))*1.1D0
C        AUL=DEXP(R1(2))

C       print*,'INITIAL VALUE USING ORDINARY EXACT CI = ',all,aul

C  COMPUTE LOWER LIMIT
       ist=1

       IF (J .EQ. 1) THEN
         ELL2=0.D0
         P_LO2=1.D0
C          PRINT*,'LOWER LIMIT =',ELL2
C          PRINT*
         GO TO 1006
      ENDIF

      IF (J .EQ. SCD) ALL=DEXP(R1(2))

C      PRINT*,'INITIAL VALUE FOR THE LOWER LIMIT = ',ALL
       CALL ITERA(ALPHA,ALL,ELL2,ist,PALPHA,IOOTO)
       P_LO2=PALPHA
C       print*,'lower limit ell2 from ITERA =',ell2
C       PRINT*

C  COMPUTE UPPER LIMIT
c       START=1.D0/AUL
1006   START=AUL
        ist=2

       IF (J .EQ. SCD) THEN
         EUL2=99999.999999
```

```
          P_UP2=1.D0
C          PRINT*,'UPPER LIMIT =',EUL2
          GO TO 1007
        ENDIF

C        print*,'start=',start
C       PRINT*,'INITIAL VALUE FOR THE UPPER LIMIT = ',AUL
        CALL ITERA(ALPHA,START,EUL2,ist,PALPHA,IOOTO)
        P_UP2=PALPHA
C        print*,'eul = ',eul
c        EUL=1.D0/EUL
C       print*,'upper limit eul2 from ITERA =',eul2

1007  CONTINUE
        IF (NCODE .EQ. 2) GO TO 1008

      PRINT*
      PRINT*,'----------------------------------------------------'
      WRITE(*,989) ELL2,EUL2
      PRINT*,'P-VALUE FOR THE LIMIT (low,up) =',P_LO2,P_UP2
      PRINT*,'----------------------------------------------------'
      PRINT*
      PRINT*,'END'
 989        FORMAT('TWO-SIDED MODIFIED EXACT CI  ',2(5X,F12.6))
       IF (NCODE .EQ. 1) GO TO 1002
C*****************************************************************
1008   CONTINUE

      IF (J .EQ. 1 .OR. J .EQ. SCD) THEN
        IF (J .EQ. 1) THEN
             OOCI(IAITN,1)=0.D0
             OOCI(IAITN,2)=DEXP(R1(2))
             OMCI(IAITN,1)=ELL1
             OMCI(IAITN,2)=EUL1
             TOCI(IAITN,1)=ELL3
             TOCI(IAITN,2)=EUL3
             TMCI(IAITN,1)=ELL2
             TMCI(IAITN,2)=EUL2
        ENDIF

        IF (J .EQ. SCD) THEN
             OOCI(IAITN,1)=DEXP(R1(2))
             OOCI(IAITN,2)=99999.99999
             OMCI(IAITN,1)=ELL1
             OMCI(IAITN,2)=EUL1
```

```
            TOCI(IAITN,1)=ELL3
            TOCI(IAITN,2)=EUL3
            TMCI(IAITN,1)=ELL2
            TMCI(IAITN,2)=EUL2
        ENDIF

       ELSE

            OOCI(IAITN,1)=DEXP(R2(2))
            OOCI(IAITN,2)=DEXP(R1(2))
            OMCI(IAITN,1)=ELL1
            OMCI(IAITN,2)=EUL1
            TOCI(IAITN,1)=ELL3
            TOCI(IAITN,2)=EUL3
            TMCI(IAITN,1)=ELL2
            TMCI(IAITN,2)=EUL2

       ENDIF

       WRITE(45,5100)IAITN,OOCI(IAITN,1),OOCI(IAITN,2)
       WRITE(46,5100)IAITN,OMCI(IAITN,1),OMCI(IAITN,2)
       WRITE(47,5100)IAITN,TOCI(IAITN,1),TOCI(IAITN,2)
       WRITE(48,5100)IAITN,TMCI(IAITN,1),TMCI(IAITN,2)
5000  CONTINUE
C*********************************************************************


C*********************************************************************
C
C           COVERAGE PROBABILITY
C
C*********************************************************************
C
      DALL=-5.51D0
      IST=1
      DO 5200 IAIN=1,1100
      DALL=DALL+0.01D0
      ALL=DEXP(DALL)
      CALL ITERA10(ALPHA,ALL,ELL2,IST,PALPHA,HYPD10)
      POOCI=0.D0
      POMCI=0.D0
      PTOCI=0.D0
      PTMCI=0.D0
      DO 5210 IAITN=1,IOTOT
      IF (ALL .GE. OOCI(IAITN,1) .AND. ALL .LE. OOCI(IAITN,2)) THEN
```

```
          POOCI=POOCI+HYPD10(IAITN,1)
        ENDIF
        IF (ALL .GE. OMCI(IAITN,1) .AND. ALL .LE. OMCI(IAITN,2)) THEN
          POMCI=POMCI+HYPD10(IAITN,1)
        ENDIF
        IF (ALL .GE. TOCI(IAITN,1) .AND. ALL .LE. TOCI(IAITN,2)) THEN
          PTOCI=PTOCI+HYPD10(IAITN,1)
        ENDIF
        IF (ALL .GE. TMCI(IAITN,1) .AND. ALL .LE. TMCI(IAITN,2)) THEN
          PTMCI=PTMCI+HYPD10(IAITN,1)
        ENDIF
 5210   CONTINUE
         COVER(IAIN,1)=DALL
         COVER(IAIN,2)=POOCI
         COVER(IAIN,3)=POMCI
         COVER(IAIN,4)=PTOCI
         COVER(IAIN,5)=PTMCI
 5200    CONTINUE

         OPEN(UNIT=50,FILE='COVER.P')
         DO 5220 IAIN=1,1100
         WRITE(50,5300)COVER(IAIN,1),COVER(IAIN,2),COVER(IAIN,3),
     1 COVER(IAIN,4),COVER(IAIN,5)
 5220    CONTINUE


        PRINT*,'END'
 5100   FORMAT(I5,2F12.6)
 5300   FORMAT(5F12.4)

 1002  END

c==============================================================
c==============================================================
      subroutine error(ierr,mxs,mxss,mxz,mxd)
      iot = 6
      if (ierr .eq. 1) then
         write(iot,10)mxs
10       format(/10x,'Error : Maximum no. of strata = ',i4)
         return
      endif
      if (ierr .eq. 2) then
         write(iot,20)mxz
20       format(/10x,'Insufficient memory : Increase size of',/,
     +                 'array HYP to be more than ',i7)
```

```
         return
      endif
      if (ierr .eq. 3) then
         write(iot,30)mxd
 30      format(/10x,'Insufficient memory : Increase size of',/,
     +                'array DS to be more than ',i7)
         return
      endif
      if (ierr .eq. 4) then
         write(iot,40)mxss
 40      format(/10x,'Error : Maximum stratum size = ',i9)
         return
      endif
      return
      end
C***************************************************
C**********
      SUBROUTINE CNV2X2(IK,MXS,MXZ,MXD,LGE,ITAB,HYP,DS,II,K1,K2,IERR,
     1                  pobsh,JCI,ODR)
C
C     CONVOLVES HYPERGEOMETRIC DISTRIBUTIONS
C     GENERATED BY SEVERAL 2X2 TABLES
C
      INTEGER ITAB(MXS,4)
      DOUBLE PRECISION HYP(0:MXZ),DS(0:1,0:MXD),SUMLG
      DOUBLE PRECISION DD1,DD2,ONE,ZERO,HYMAX,DSMX,LGE,EL
      DOUBLE PRECISION ZLOG,ZEXP,X
      DOUBLE PRECISION hypsum(1000),hypobs(1000),pobsh,hypd(1000,0:2000)
      DOUBLE PRECISION DENO1,POBSH1,PEXIMP,PEX,ODR,PSI

      integer infhyl(1000),infhyu(1000)
      COMMON/CI2/hypd,infhyl,infhyu,POBSH1,PEXIMP,PEX

      DATA ONE,ZERO /1.0D+00,0.0D+00/
C
      ZLOG(X) = DLOG(X)
      ZEXP(X) = DEXP(X)
C
C     CHECK INPUT PARAMETERS
C
      IF (IK .GT. MXS) IERR=1
      K1 = 0
      DO 1 I=1,IK
         IMM = ITAB(I,1) + ITAB(I,2)
         INN = ITAB(I,3) + ITAB(I,4)
```

```
          ITT = ITAB(I,1) + ITAB(I,3)
C
C      LOWER AND UPPER LIMITS FOR STRATUM DISTRIBUTION
C
          IF (ITT .GT. INN) THEN
             IL1 = ITT-INN
          ELSE
             IL1 = 0
          ENDIF
          IF (ITT .LT. IMM) THEN
             IL2 = ITT
          ELSE
             IL2 = IMM
          ENDIF
          ITT = IL2 - IL1
          IF (ITT .GT. MXZ) IERR = 2
          K1 = K1 + IL1
    1 CONTINUE
      IF (IERR .GT. 0) RETURN
C
      DD1 = 10*ONE
      EL  = LGE*ZLOG(DD1)
C
C     INITIALISE AND SET LOG-SCALE INDICATOR
C
      II = 0
      JJ = 1
      IR = 0
      DS(0,0) = ONE/DEXP(EL)
      DSMX = ZERO - EL
      ILS = 0
C
C     FOR STRATA=1,...,IK, COMPUTE HYPERGEOMETRIC DISTRIBUTION
C     AND PERFORM CONVOLUTION IN A RECURSIVE FASHION
C
      DO 13 I=1,IK
         IMM = ITAB(I,1) + ITAB(I,2)
         INN = ITAB(I,3) + ITAB(I,4)
         ITT = ITAB(I,1) + ITAB(I,3)
C
C     LOWER AND UPPER LIMITS FOR CONVOLUTION
C
         IF (ITT .GT. INN) THEN
            IL1 = ITT-INN
         ELSE
```

```
            IL1 = 0
         ENDIF
         IF (ITT .LT. IMM) THEN
            IL2 = ITT
         ELSE
            IL2 = IMM
         ENDIF
         IL2 = IL2 - IL1
         K2 = IR + IL2
         IF (K2 .GT. MXD) THEN
            IERR = 3
            RETURN
         ENDIF
C
C     COMPUTE STRATUM DISTRIBUTION ON LOG-SCALE
C
         HYP(0) = ZERO
         DO 2 J=1,IL2
            DD1 = DBLE(FLOAT(IMM-J-IL1+1))*DBLE(FLOAT(ITT-J-IL1+1))
            DD2 = DBLE(FLOAT(J+IL1))*DBLE(FLOAT(INN-ITT+J+IL1))
            HYP(J) = HYP(J-1) + ZLOG(DD1/DD2)
C            print*,'data ?',j,HYP(J),zexp(HYP(J)),j+IL1
    2    CONTINUE
         IF (ILS .EQ. 1) GOTO 9
C
C     GET MAXIMUM HYPERGEOMETRIC COEFFICIENT ON LOG-SCALE AND
C     CHECK FOR POTENTIAL OVERFLOW IN STRATUM DISTRIBUTION
C
         AM = (1.0 + FLOAT(ITT))/(1.0 + FLOAT(INN+1)/FLOAT(IMM+1))
         IAM = IFIX(AM) - IL1
         IF (HYP(0) .GT. HYP(IL2)) THEN
            DO 3 J=0,IL2
               HYP(J) = HYP(J) - HYP(IL2)
C               print*,'j,hyp(j),zexp(HYP(J))',j,hyp(j),zexp(HYP(J))
    3       CONTINUE
         ENDIF
         HYMAX = HYP(IAM)
         IF (HYMAX .GT. EL) THEN
            ILS = 1
            print*,'ILS=1 in CI'
            GOTO 7
         ENDIF
C
C     CHECK FOR POTENTIAL OVERFLOW IN THE ITH CONVOLUTION
C
```

```
          IF (IL2 .LT. IR) THEN
             IXX = IL2 + 1
          ELSE
             IXX = IR + 1
          ENDIF
C          IXX = (IL2 + 1)*(IR + 1)
          DD1 = DBLE(FLOAT(IXX))
          DD1 = ZLOG(DD1)
          DSMX = DSMX + HYMAX + DD1

C          WRITE(*,*)I,DSMX,HYMAX,DD1
          IF (DSMX .GE. EL - ONE) THEN
             ILS = 1
             print*,'ILS=1 in CI'
             GOTO 7
          ENDIF
C
C     CONVERT STRATUM DISTRIBUTION TO NATURAL SCALE
C
          hypsum(i)=0.d0
C          PRINT*,'ODR,JCI',ODR,JCI

          DO 4 J=0,IL2
      IF (JCI .EQ. 1) THEN
        HYP(J) = ZEXP(HYP(J))*ODR**(J+IL1)
        GO TO 9999
      ENDIF
            HYP(J) = ZEXP(HYP(J))
C            ihyp(i,j)=j
9999        hypd(i,j)=hyp(j)
c   hypd(i,j) is hypergeometric prob dist for each stratum.

            hypsum(i)=hypsum(i)+hyp(j)
C          print*,'HYP(J)',J, HYP(J),hypsum(i)
    4     CONTINUE
          infhyl(i)=il1
          infhyu(i)=il2
c   infhyu(i) is the no. of possible tables for the fixed 2x2 tables.

          ikim=itab(i,1)-IL1
          hypobs(i)=hyp(ikim)
C   HYPOBS(I) IS THE NUMERATOR OF PROB FOR EACH STRATUM.
C     PRINT*,'STRATUM PROB',I,hyp(ikim),hypsum(i),hypobs(i)/HYPSUM(I)
C
C     PERFORM CONVOLUTION ON NATURAL SCALE
```

```
C
         DSMX = ZERO - EL
         DO 6 J=0,K2
            IF (J .GT. IL2) THEN
               IH1 = J-IL2
            ELSE
               IH1 = 0
            ENDIF
            IF (J .LT. IR) THEN
               IH2 = J
            ELSE
               IH2 = IR
            ENDIF
            DS(JJ,J) = DS(II,IH1)*HYP(J-IH1)
            DO 5 JR=IH1+1,IH2
               DD1 = DS(II,JR)*HYP(J-JR)
               DS(JJ,J) = DS(JJ,J) + DD1
    5       CONTINUE
            DD1 = ZLOG(DS(JJ,J))
            IF (DD1 .GT. DSMX) DSMX = DD1
    6    CONTINUE
         GOTO 12
    7    CONTINUE
C
C     CONVERT (I-1)TH CONVOLVED DISTRIBUTION TO LOG SCALE
C
         DO 8 KK=0,IR
            DS(II,KK) = EL + ZLOG(DS(II,KK))
    8    CONTINUE
C
C     PERFORM CONVOLUTION ON LOGARITHMIC SCALE
C
    9    CONTINUE
c----------------------------------------
C
         if (ils .eq. 1) then
         hypsum(i)=0.d0
C         PRINT*,'ODR,JCI',ODR,JCI

         DO 4444 J=0,IL2
      IF (JCI .EQ. 1) THEN
        HYP(J) = ZEXP(HYP(J))*ODR**(J+IL1)
        GO TO 9998
      ENDIF
            HYP(J) = ZEXP(HYP(J))
```

```
C           ihyp(i,j)=j
9998        hypd(i,j)=hyp(j)
c   hypd(i,j) is hypergeometric prob dist for each stratum.

            hypsum(i)=hypsum(i)+hyp(j)
C          print*,'HYP(J)',J, HYP(J),hypsum(i)
 4444    CONTINUE
         infhyl(i)=il1
         infhyu(i)=il2
c   infhyu(i) is the no. of possible tables for the fixed 2x2 tables.

         ikim=itab(i,1)-IL1
         hypobs(i)=hyp(ikim)
C   HYPOBS(I) IS THE NUMERATOR OF PROB FOR EACH STRATUM.
         endif
c-----------------------------------------
         DO 11 J=0,K2
            IF (J .GT. IL2) THEN
               IH1 = J-IL2
            ELSE
               IH1 = 0
            ENDIF
            IF (J .LT. IR) THEN
               IH2 = J
            ELSE
               IH2 = IR
            ENDIF
            DS(JJ,J) = DS(II,IH1) + HYP(J-IH1)
            DO 10 JR=IH1+1,IH2
               DD1 = DS(II,JR) + HYP(J-JR)
               DD2 = DS(JJ,J)
               DS(JJ,J) = SUMLG(DD1,DD2)
   10       CONTINUE
   11    CONTINUE
C
C     RESET FOR NEXT STEP
C
   12    II = JJ
         JJ = 1 - II
         IR = K2
C
C     DELETE THE NEXT TWO STATEMENTS FROM
C     THE PUBLISHED ALGORITHM
C
C         write(*,95)i,ils
```

```
95        format(/10x,'stratum no. ',i3,4x,'scale =',i2)
C
C
   13 CONTINUE

C  ADDED BY DONGUK KIM
      POBSH=1.D0
      DO 20 I=1,IK
   20 POBSH=POBSH*HYPOBS(I)
      POBSH1=POBSH
C  POBSH IS OBSERVED VALUE AND PROB IS POBSH/DENO1

      DENO1=1.D0
      DO 21 I=1,IK
   21 DENO1=DENO1*HYPSUM(I)
      POBSH=POBSH/DENO1

C       PRINT*,'PROB OF OBSERVED TABLE=',POBSH
C       PRINT*,'OBSERVED VALUE FOR PROB=',POBSH1

C
C     NORMALISE FINAL DISTRIBUTION
C
      IF (ILS .NE. 1) THEN
         DO 14 I=0,K2
            DS(II,I) = ZLOG(DS(II,I))
   14    CONTINUE
      ENDIF
      DSMX = DS(II,0)
      DO 15 I=1,K2
         DD1 = DS(II,I)
         DD2 = SUMLG(DSMX,DD1)
         DSMX = DD2
   15 CONTINUE
      DO 16 I=0,K2
         DS(II,I) = DS(II,I) - DSMX
   16 CONTINUE
      K2 = K1 + K2


c   Added by DONGUK KIM
C
C  CALCULATES OBSERVED POSITION IN SAMPLE SPACE
C  J   - POSITION
C  SCD - SIZE COND. SAMPLE SPACE
```

```
      ISUMA=0
      DO 180 I=1,IK
         ISUMA = ISUMA + ITAB(I,1)
 180  CONTINUE

      J = ISUMA - K1 + 1
      SCD = K2 - K1 + 1
      IF (JCI .EQ. 1) GO TO 999

      JCIO=0
C      IF (JCI .EQ. 1) GO TO 999
      CALL IMPROV(ik,itab,hypd,infhyl,infhyu,POBSH1,PEXIMP,PEX,JCIO,PSI)

999   RETURN
      END

C*********************************************************************



C*********************************************************************
C   ADDED BY DONGUK KIM DEC.1, 1992
C   ENUMERATE ALL POSSIBLE TABLES WITH GIVEN 2x2 MARGINS, AND
C   COMPUTE IMPROVED EXACT AND EXACT UPPER AND LOWER TAIL PROBABILITY.

      SUBROUTINE IMPROV(ik,itab,hypd,infhyl,infhyu,POBSH1,PEXIMP,
     1 PEX,JCIO,PSI)
      PARAMETER(MAXT=270000)
      DOUBLE PRECISION HYPD(1000,0:2000),HYPSUM(1000),HYPD1(270000,1)
      DOUBLE PRECISION DENO,TS,POBSH1,PTOBS1,PTOBS2,PEXIMP,PEX
      DOUBLE PRECISION PEXIML,PEXIMU,PEXL,PEXU,HYPD2(270000,20)
      DOUBLE PRECISION PEXP1,PEXP2,X,PDN1,PDN2
      DOUBLE PRECISION PVDIS(270000,5),PDT(500,3),PSI

      DOUBLE PRECISION CHI(270000),CHI1(270000),CMH,CHIOBS,G2(270000)
      DOUBLE PRECISION G,GOBS,CUP,FIT(1000,2,2)

C PDT(500,3):Pr(P-value<=x) for two P-values.

      INTEGER ITAB(1000,4),INFHYL(1000),INFHYU(1000)
      INTEGER INUM(270000,20),INUM1(270000,1),IOTOT
      INTEGER NIK(2,1000),NJK(2,1000),NTOT(1000),MATRIX(1000,2,2)

      LOGICAL ISEA
```

```
      COMMON /DKIM/ DENO,ITOT,ISUML,INUM,HYPD2,INUM1,HYPD1
c      COMMON /DKIM1/ POBSH1,ISUMTS,IK
      COMMON /DKIM1/ ISUMTS
      COMMON /CH/ NIK,NJK,NTOT
      COMMON /ART/ IOTOT
      COMMON /CH1/ CHI,CHIOBS


C  HYPD1 HAS PROB HAVING T .GE. T_OBS.
C  INUM1 HAS VALUE IN EACH STRATUM HAVING T .GE. T_OBS.
C  THE MAXIMUM NO OF TABLES FOR T .GE. T_OBS IS ALLOWED TO BE 5500.
C  IF IT IS GREATER THAN 5500, MAXT IN HYPD1(MAXT,1000) AND
C  INUM1(MAXT,1000) SHOULD BE INCREASED.

      DENO=1.D0
      ISUML=0
      ISUMU=0

      DO 100 I=1,IK
      HYPSUM(I)=0.D0
      IJ=INFHYU(I)
C      print*,'I,IL1,IL2,HYPSUM(I)',I,INFHYL(I),INFHYU(I),HYPSUM(I)
C  IJ IS IL2 FOR EACH STRATUM
      DO 110 J=0,IJ
110   HYPSUM(I)=HYPSUM(I)+HYPD(I,J)
      ISUML=ISUML+INFHYL(I)
      ISUMU=ISUMU+INFHYU(I)
      DENO=DENO*HYPSUM(I)
C   DENO IS DENOMINATOR OF THE HYPERGEO. PROB. DIST.
100   CONTINUE

C      PRINT*,'DENO=',DENO

      ISUMA=0
      DO 120 I=1,IK
      ISUMA=ISUMA+ITAB(I,1)
120   CONTINUE

      ISUMTS=ISUMA-ISUML
C      PRINT*,'ISUMA,ISUMTS',ISUMA,ISUMTS
C  ISUMTS IS OBSERVED TEST STATISTICS WHICH WILL BE USED.

      ITOT=1
      DO 130 I=1,IK
```

```
 130   ITOT=ITOT*(INFHYU(I)+1)
       IOTOT=ITOT

 C      PRINT*,'NO. OF POSSIBLE TABLES = ',ITOT
 C   ITOT IS TOTAL NUMBER OF ENUMERATION FOR THE TABLES.

 C----------------------------------------------------------
       IF (ISEA) GO TO 315
 C
 C     SET ISEA FOR THE SUBSEQUENT CALLS
 C
       ISEA=.TRUE.

 C  TO MAKE INUM

       ICOUNT=0
       NUM=1

       DO 210 K1=1,IK-1
       DO 220 K2=K1+1,IK
 220   NUM=NUM*(INFHYU(K2)+1)
 C   NUM IS NUMBER OF REPLICATES.

       K=K1
       IN=INFHYU(K)
 99    DO 230 K3=1,NUM
       ICOUNT=ICOUNT+1
       INUM(ICOUNT,K)=IN
 230   CONTINUE

       IF (IN .GT. 0) THEN
        IN=IN-1
        GO TO 99
       ENDIF

       IF (ICOUNT .LT. ITOT) THEN
        IN=INFHYU(K)
        GO TO 99
       ELSE
        ICOUNT=0
        NUM=1
       ENDIF
 210   CONTINUE

 C  THE LAST STRATUM (i.e., LAST COLUMN IN ARRAY)
```

```
      ICOUNT=0
      K=IK
      IN=INFHYU(K)
 250  ICOUNT=ICOUNT+1
      INUM(ICOUNT,K)=IN

      IF (IN .GT. 0) THEN
       IN=IN-1
       GO TO 250
      ENDIF

      IF (ICOUNT .LT. ITOT) THEN
       IN=INFHYU(K)
       GO TO 250
      ENDIF
C      PRINT*,'ICOUNT=',ICOUNT
C      PRINT*,'ITOT=',ITOT

      DO 300 I=1,ITOT
      ITS=0
      DO 310 J=1,IK
310   ITS=ITS+INUM(I,J)
      INUM(I,IK+1)=ITS
300   CONTINUE
315   CONTINUE
C-----------------------------------------------------------

C      PRINT*,'DISPLAY ALL POSSIBLE TABLES (Y=1,N=0) ?'
C      READ(*,*)IDIS
      IDIS=0

      IF (IDIS .EQ.1) THEN
         PRINT*,'INPUT NO. OF INCREMENT :'
C         READ(*,*)INCR
      INCR=1

         PRINT*,'ENUMERATION'
         DO 320 I=1,ITOT,INCR
         PRINT*,I,(INUM(I,J)+INFHYL(J),J=1,IK),INUM(I,IK+1)+ISUML
320      CONTINUE
      ENDIF
C      PRINT*

C  COMPUTATION OF PROB OF ALL RANDOM TABLES
      DO 350 K=1,IK
```

```
      DO 355 I=1,ITOT
      IA=INUM(I,K)
      HYPD2(I,K)=HYPD(K,IA)
 355  CONTINUE
 350  CONTINUE

      DO 360 I=1,ITOT
      TS=1.D0
      DO 365 J=1,IK
 365  TS=TS*HYPD2(I,J)
      HYPD2(I,IK+1)=TS
 360  CONTINUE

C    PRINT PROB OF ALL RANDOM TABLE
      TS=0.D0
       DO 367 I=1,ITOT
       TS=TS+HYPD2(I,IK+1)
 367   CONTINUE
C       PRINT*,'SUM OF VALUE, DENO, PROB=',TS,DENO,TS/DENO
C       PRINT*

C      PRINT*,'PROB OF ALL RANDOM TABLES (Y=1,N=0) ?'
C      READ(*,*)IDSA
      IDSA=0

      IF (IDSA .EQ. 1) THEN
       PRINT*,'ENUMERATION OF PROB OF ALL RANDOM TABLES'
       DO 370 I=1,ITOT,INCR
       PRINT*,I,(sngl(HYPD2(I,J)),J=1,IK),sngl(HYPD2(I,IK+1)/DENO)
 370   CONTINUE
      ENDIF
C      print*


C--------------------------------
C
C  GENERATE ALL POSSIBLE RANDOM TABLES
C
C--------------------------------
C      WRITE(*,70010)
C70010 FORMAT(/,'PRINT X^2 AND G^2 FOR ALL RANDOM TABLES ? (Y=1,N=0)')
C      READ(*,*)IX2G2
       IX2G2=0
C       PRINT*,'IX2G2=0'
```

```
      NROW=2
      NCOL=2
      NSTM=IK
      CUP=0.D0

C  IPF IS CALLED JUST ONE TIME WITHIN FIXED OR.
      IF (JCIO .NE. 0) THEN
            DO 375 K=1,IK
            MATRIX(K,1,1)=ITAB(K,1)
            MATRIX(K,1,2)=ITAB(K,2)
            MATRIX(K,2,1)=ITAB(K,3)
            MATRIX(K,2,2)=ITAB(K,4)
375      CONTINUE
      CALL IPF(PSI,IK,MATRIX,FIT)
      ENDIF

      DO 384 I=1,ITOT
      DO 386 K=1,IK
      MATRIX(K,1,1)=INUM(I,K)+INFHYL(K)
      MATRIX(K,1,2)=NIK(1,K)-MATRIX(K,1,1)
      MATRIX(K,2,1)=NJK(1,K)-MATRIX(K,1,1)
      MATRIX(K,2,2)=NTOT(K)-(MATRIX(K,1,1)+MATRIX(K,1,2)+MATRIX(K,2,1))

      IF (IX2G2 .EQ. 1) THEN
      WRITE(*,70000)I,K,MATRIX(K,1,1),MATRIX(K,1,2),MATRIX(K,2,1),
     1 MATRIX(K,2,2)
C      WRITE(*,70001)K,NIK(1,K),NIK(2,K),NJK(1,K),NJK(2,K),NTOT(K)
70000  FORMAT(2I5,' : ',4I10)
70001  FORMAT('TOTAL',6I10)
      ENDIF

386   CONTINUE

C      IF (JCIO .NE. 0) CALL IPF(PSI,IK,MATRIX,FIT)

      CALL CMHNN1(NROW,NCOL,NSTM,NIK,NJK,NTOT,MATRIX,CMH,G,
     1  JCIO,FIT)

      CHI(I)=CMH
      G2(I)=G
      CUP=CUP+HYPD2(I,IK+1)/DENO

      IF (IX2G2 .EQ. 1) THEN
      WRITE(*,70002)I,HYPD2(I,IK+1)/DENO,CHI(I),G2(I),CUP
70002 FORMAT('NO., Pr(T), X^2, G^2 = ',I5,4F14.7,/)
```

```
      ENDIF

 384   CONTINUE
 C   COMPUTE THE OBSERVED CHI-SQUARED STATISTIC : CHIOBS
       IF (IX2G2 .EQ. 1) THEN
       PRINT*
       PRINT*,'OBSERVED DATA'
       ENDIF

       DO 390 K=1,IK
       MATRIX(K,1,1)=ITAB(K,1)
       MATRIX(K,1,2)=ITAB(K,2)
       MATRIX(K,2,1)=ITAB(K,3)
       MATRIX(K,2,2)=ITAB(K,4)

       IF (IX2G2 .EQ. 1) THEN
       WRITE(*,70004)K,ITAB(K,1),ITAB(K,2),ITAB(K,3),
      1 ITAB(K,4)
70004 FORMAT(5X,I5,' : ',4I10)
C       WRITE(*,70001)K,NIK(1,K),NIK(2,K),NJK(1,K),NJK(2,K),NTOT(K)
       ENDIF

390    CONTINUE
       CALL CMHNN1(NROW,NCOL,NSTM,NIK,NJK,NTOT,MATRIX,CMH,G,
      1   JCIO,FIT)

       CHIOBS=CMH
       GOBS=G
       IF (IX2G2 .EQ.1) THEN
       WRITE(*,70003)POBSH1/DENO,CHIOBS,GOBS
70003  FORMAT('OBSERVED Pr(t_o), X^2, G^2 = ',3F14.7,/,/)
       ENDIF
C-------------------------------------------------------------------

395    IF (JCIO .EQ. 3) GO TO 1000

C-------------------------------------------------------------------
C  UPPER TAIL
C TO MAKE INUM1

C KIM_9.F

       IF (JCIO .EQ. 1) GO TO 666
       ICOUNT1=0
       DO 400 I=1,ITOT
```

```
        IF (INUM(I,IK+1) .GE. ISUMTS) THEN
         ICOUNT1=ICOUNT1+1
         INUM1(ICOUNT1,1)=INUM(I,IK+1)
         HYPD1(ICOUNT1,1)=HYPD2(I,IK+1)
         CHI1(ICOUNT1)=CHI(I)


          IF (ICOUNT1 .GE. MAXT) THEN
             PRINT*,'INCREASE ARRAY INUM1,HYPD1 IN SUBROUTINE IMPRIV'
             print*,'icount,icount1',icount,icount1
             go to 1000
          ENDIF

       ENDIF
 400   CONTINUE

       IF (JCIO .EQ. 0) THEN
 C        PRINT*,'NO OF TABLES FOR T .GE. T_OBS =',ICOUNT1
       ENDIF

 C      PRINT*
 C      PRINT*,'DISPLAY ALL TABLES FOR T .GE. T_OBS (Y=1,N=0) ?'
 C      READ(*,*)IDIS1
       IDIS1=0

       IF (IDIS1 .EQ. 1) THEN
c*************
c         PRINT*,'INPUT NO. OF INCREMENT :'
c         READ(*,*)INCR1
         INCR1=1
c*************

          PRINT*,'ENUMERATION FOR THOSE TABLES HAVING T .GE. T_OBS'
          DO 420 I=1,ICOUNT1,INCR1
          PRINT*,I,INUM1(I,1)+ISUML
420       CONTINUE
       ENDIF
C       PRINT*

C  COMPUTATION OF PROB OF OBSERVING OBSERVED AND RANDOM TABLES

C       PRINT*,'PROB FOR THOSE TABLES HAVING T .GE. T_OBS (Y=1,N=0) ?'
C       READ(*,*)IDS2
       IDS2=0
```

```
      IF (IDS2 .EQ. 1) THEN
       PRINT*,'ENUMERATION OF PROB FOR THOSE TABLES HAVING T .GE. T_OBS'
       DO 550 I=1,ICOUNT1,INCR
       PRINT*,I,sngl(HYPD1(I,1)/DENO)
550    CONTINUE
      ENDIF
C      print*

      PTOBS1=0.D0
C      PRINT*,'DISPLAY UPPER TAIL IMPROV. PROB (Y=1,N=0) ?'
C      READ(*,*)IDS3
      IDS3=0

      IF (IDS3 .EQ. 1) WRITE(*,888)
c      WRITE(*,888)
      DO 560 I=1,ICOUNT1
      IF (INUM1(I,1) .GT. ISUMTS .OR. INUM1(I,1) .EQ. ISUMTS
     1    .AND. CHI1(I) .GE. CHIOBS) THEN
C     1    .AND. HYPD1(I,1) .LE. POBSH1) THEN
        PTOBS1=PTOBS1+HYPD1(I,1)

        if (IDS3 .NE. 1) GO TO 555
        WRITE(*,900)INUM1(I,1)+ISUML,ISUMTS+ISUML,
     2   HYPD1(I,1)/DENO,POBSH1/DENO,PTOBS1/DENO
900     FORMAT(2(1X,I7),3(1X,F12.6))
555   ENDIF
560   CONTINUE
c      ENDIF
C      print*

      PEXIMU=PTOBS1/DENO
C  PEXIMPU IS IMPROVED UPPER TAIL EXACT PROB.

      PTOBS2=0.D0
C      PRINT*,'DISPLAY UPPER TAIL PROB (Y=1,N=0) ?'
C      READ(*,*)IDS4
      IDS4=0

      IF (IDS4 .EQ. 1) WRITE(*,889)
C      WRITE(*,889)
      DO 570 I=1,ICOUNT1
      IF (INUM1(I,1) .GE. ISUMTS) THEN
        PTOBS2=PTOBS2+HYPD1(I,1)
        IF (IDS4 .NE. 1) GO TO 565
        WRITE(*,900)INUM1(I,1)+ISUML,ISUMTS+ISUML,
```

```
     3   HYPD1(I,1)/DENO,POBSH1/DENO,PTOBS2/DENO
C        PRINT*,INUM1(I,1),ISUMTS,SNGL(HYPD1(I,1)/DENO),
C     3          SNGL(POBSH1/DENO),SNGL(PTOBS2/DENO)
565   ENDIF
570   CONTINUE
C      ENDIF
C      print*

      PEXU=PTOBS2/DENO
C  PEXU  IS UPPER TAIL EXACT PROB.

C      PRINT*,'IMPROVED P_EXACT =',PEXIMP
C      PRINT*,'    P_EXACT      =',PEX
c      PRINT*,'PROB. OF OBSERVED TABLES =',POBSH1/DENO
C      WRITE(*,901)PEXIMU
C      WRITE(*,902)PEXU
C      PRINT*
C      PRINT*,'PROB. OF OBSERVED TABLES =',POBSH1/DENO
C      PRINT*

      IF (JCIO .EQ. 2) THEN
         PEXIMP=PEXIMU
         PEX=PEXU
         GO TO 1000
      ENDIF

888   FORMAT('        T      T_obs     Pr(Ta)      pr(Ta_obs)    P(T>T_obs+
     1order)')
889   FORMAT('        T      T_obs     Pr(Ta)      pr(Ta_obs)    P(T>=T_obs)
     2')

901   FORMAT('IMPROVED UPPER P_EXACT =',3X,F12.6)
902   FORMAT('   UPPER    P_EXACT     =',3X,F12.6)

c--------------------------------------------------------------------

C--------------------------------------------------------------------
C  LOWER TAIL
C TO MAKE INUM1
C  KIM_9.F

666   ICOUNT1=0
      DO 600 I=1,ITOT
      IF (INUM(I,IK+1) .LE. ISUMTS) THEN
       ICOUNT1=ICOUNT1+1
```

```
        INUM1(ICOUNT1,1)=INUM(I,IK+1)
        HYPD1(ICOUNT1,1)=HYPD2(I,IK+1)
        CHI1(ICOUNT1)=CHI(I)


         IF (ICOUNT1 .GE. MAXT) THEN
            PRINT*,'INCREASE ARRAY INUM1,HYPD1 IN SUBROUTINE IMPRIV'
            print*,'icount,icount1',icount,icount1
            go to 1000
         ENDIF

      ENDIF
600   CONTINUE

      IF (JCIO .EQ. 0) THEN
C        PRINT*,'NO OF TABLES FOR T .LE. T_OBS =',ICOUNT1
      ENDIF

C      PRINT*
C      PRINT*,'DISPLAY ALL TABLES FOR T .LE. T_OBS (Y=1,N=0) ?'
C      READ(*,*)IDIS1
      IDIS1=0

      IF (IDIS1 .EQ. 1) THEN
C*************
c         PRINT*,'INPUT NO. OF INCREMENT :'
c         READ(*,*)INCR1
         INCR1=1
c**************
         PRINT*,'ENUMERATION FOR THOSE TABLES HAVING T .LE. T_OBS'
         DO 620 I=1,ICOUNT1,INCR1
         PRINT*,I,INUM1(I,1)+ISUML
620      CONTINUE
      ENDIF
C      PRINT*

C  COMPUTATION OF PROB OF OBSERVING OBSERVED AND RANDOM TABLES

C      PRINT*,'PROB FOR THOSE TABLES HAVING T .LE. T_OBS (Y=1,N=0) ?'
C      READ(*,*)IDS2
      IDS2=0

      IF (IDS2 .EQ. 1) THEN
       PRINT*,'ENUMERATION OF PROB FOR THOSE TABLES HAVING T .LE. T_OBS'
       DO 750 I=1,ICOUNT1,INCR
```

```
        PRINT*,I,sngl(HYPD1(I,1)/DENO)
750     CONTINUE
      ENDIF
C      print*

      PTOBS1=0.D0
C      PRINT*,'DISPLAY LOWER TAIL IMPROV. PROB (Y=1,N=0) ?'
C      READ(*,*)IDS3
      IDS3=0

      IF (IDS3 .EQ. 1) WRITE(*,890)
c      WRITE(*,890)
      DO 760 I=1,ICOUNT1
      IF (INUM1(I,1) .LT. ISUMTS .OR. INUM1(I,1) .EQ. ISUMTS
     1    .AND. CHI1(I) .GE. CHIOBS) THEN
C     1    .AND. HYPD1(I,1) .LE. POBSH1) THEN

        PTOBS1=PTOBS1+HYPD1(I,1)
        if (IDS3 .NE. 1) GO TO 755
        WRITE(*,900)INUM1(I,1)+ISUML,ISUMTS+ISUML,
     2   HYPD1(I,1)/DENO,POBSH1/DENO,PTOBS1/DENO
755   ENDIF
760   CONTINUE
c      ENDIF
C      print*

      PEXIML=PTOBS1/DENO
C  PEXIMPL IS IMPROVED LOWER TAIL EXACT PROB.

      PTOBS2=0.D0
C      PRINT*,'DISPLAY LOWER TAIL PROB (Y=1,N=0) ?'
C      READ(*,*)IDS4
      IDS4=0

      IF (IDS4 .EQ. 1) WRITE(*,891)
C      WRITE(*,891)
      DO 770 I=1,ICOUNT1
      IF (INUM1(I,1) .LE. ISUMTS) THEN
        PTOBS2=PTOBS2+HYPD1(I,1)
        IF (IDS4 .NE. 1) GO TO 765
        WRITE(*,900)INUM1(I,1)+ISUML,ISUMTS+ISUML,
     3   HYPD1(I,1)/DENO,POBSH1/DENO,PTOBS2/DENO
C        PRINT*,INUM1(I,1),ISUMTS,SNGL(HYPD1(I,1)/DENO),
C     3          SNGL(POBSH1/DENO),SNGL(PTOBS2/DENO)
765   ENDIF
```

```
770   CONTINUE
C      ENDIF
C      print*

      PEXL=PTOBS2/DENO
C  PEXL IS LOWER TAIL EXACT PROB.

C      PRINT*,'IMPROVED LOWER P_EXACT =',PEXIML
C      PRINT*,'   LOWER  P_EXACT      =',PEXL
c      PRINT*,'PROB. OF OBSERVED TABLES =',POBSH1/DENO
C      WRITE(*,903)PEXIML
C      WRITE(*,904)PEXL
C      PRINT*
C      PRINT*,'PROB. OF OBSERVED TABLES =',POBSH1/DENO
C      PRINT*

      IF (JCIO .EQ. 1) THEN
           PEXIMP=PEXIML
           PEX=PEXL
           GO TO 1000
      ENDIF


890   FORMAT('       T      T_obs    Pr(Ta)     pr(Ta_obs)   P(T<T_obs+
     1order)')
891   FORMAT('       T      T_obs    Pr(Ta)     pr(Ta_obs)   P(T<=T_obs)
     2')

903   FORMAT('IMPROVED LOWER P_EXACT =',3X,F12.6)
904   FORMAT('   LOWER   P_EXACT     =',3X,F12.6)

c------------------------------------------------------------------------

      IF (PEXIML .GT. PEXIMU) PEXIML=PEXIMU
      PEXIMP=PEXIML

      IF (PEXL .GT. PEXU) PEXL=PEXU
      PEX=PEXL

C      PRINT*
C      PRINT*,'-----------------------------------------------'
C      WRITE(*,906)PEXIMP
C      WRITE(*,907)PEX
C      PRINT*
C      PRINT*,'PROB. OF OBSERVED TABLES =',POBSH1/DENO
```

```
C      PRINT*,'------------------------------------------------'
C      PRINT*

906   FORMAT('IMPROVED P_EXACT =',3X,F12.6)
C906   FORMAT('  MID P P_EXACT  =',3X,F12.6)
907   FORMAT('      P_EXACT     =',3X,F12.6)

C=========================================
C      WRITE(*,1200)
1200  FORMAT('EXPECTED VALUE OF P_VALUES (Y=1,N=0) ?')
C      READ(*,*)IEP
      IEP=0

      IF (IEP .NE. 1) GO TO 1000
1215  WRITE(*,1210)
1210  FORMAT(/,'ENTER CODE : 1. LOWER TAIL ONLY',/,
     1     '               2. UPPER TAIL ONLY')
C      READ(*,*)IUL

C-----------------------------------------------------------
C  SELECT  LOWERTAIL OR UPPER TAIL
      IUL=1
C-----------------------------------------------------------

      PRINT*,'CODE =',IUL
      PRINT*

      IF (IUL .NE. 1 .AND. IUL .NE. 2) GO TO 1215
      PEXP1=0.D0
      PEXP2=0.D0

      DO 1220 I=1,ITOT

      INO=I
      ISUMTS=INUM(I,IK+1)
      POBSH1=HYPD2(I,IK+1)
C      PRINT*,'NO. =',I,' T =',ISUMTS+ISUML,' Pr(Ta) =',POBSH1/DENO
      CALL MEANP(INO,IK,HYPD,ISUMTS,POBSH1,IUL,PEXIMP,PEX)
      PEXP1=PEXP1+PEXIMP*POBSH1/DENO
      PEXP2=PEXP2+PEX*POBSH1/DENO

      PVDIS(I,1)=DBLE(INO)
      PVDIS(I,2)=DBLE(ISUMTS+ISUML)
      PVDIS(I,3)=POBSH1
      PVDIS(I,4)=PEXIMP
```

```
      PVDIS(I,5)=PEX
 1220 CONTINUE

      PRINT*
      PRINT*,'-----------------------------------------------'
      WRITE(*,1230) PEXP1
      WRITE(*,1240) PEXP2
      PRINT*,'-----------------------------------------------'
      PRINT*
      PRINT*

      OPEN(UNIT=31,FILE='mean.out')


      WRITE(31,1250) PEXP1,PEXP2
 1250  FORMAT('E_P_improved = ',F12.6,5x,'E_P_ordinary = ',F12.6)

 1230  FORMAT('MEAN OF IMPROVED EXACT P_VALUES =',F12.6)
 1240  FORMAT('MEAN OF STANDARD EXACT P_VALUES =',F12.6)

c=========================================
      WRITE(*,1300)
 1300 FORMAT('THE CDF OF TWO P-VALUES (Y=1,N=0) ?',/,
     1  ' i.e., Pr(P-value <= x ,0<x<1) ?')
      PRINT*
C      READ(*,*)IDP
      IDP=0
      PRINT*,'CDF=0'
      PRINT*

      IF (IDP .NE. 1) GO TO 1000

      DO 1310 IX=1,500
      X=DBLE(IX)/500.D0
      PDN1=0.D0
      PDN2=0.D0
      DO 1320 I=1,ITOT
      IF (PVDIS(I,4) .LE. X) PDN1=PDN1+PVDIS(I,3)
      IF (PVDIS(I,5) .LE. X) PDN2=PDN2+PVDIS(I,3)
 1320 CONTINUE
      PDT(IX,1)=X
      PDT(IX,2)=PDN1
      PDT(IX,3)=PDN2
 1310 CONTINUE
C  PDT(IX,2)/DENO FOR IMPROVED AND  PDT(IX,3)/DENO FOR ORDINARY
```

```
C  IS Pr(P-value<=x) for x=PDT(IX,1).


      OPEN(UNIT=32,FILE='pcdf.out')
      DO 1330 IX=1,500
      WRITE(32,1340) PDT(IX,1),PDT(IX,2)/DENO,PDT(IX,3)/DENO
1330  CONTINUE
1340  FORMAT(F12.6,1X,F12.6,1X,F12.6)

1000  RETURN
      END

C***********************************************************************

C*********************************************************************
C   FROM ALL POSSIBLE TABLES WITH GIVEN 2x2 MARGINS
C   COMPUTE ALL POSSIBLE STANDARD AND IMPROVED EXACT
C   UPPER OR LOWER TAIL PROBABILITY.
C  INUM,  HYPD2 IS FOR ALL ENUMERATION.
C  INUM1, HYPD1 IS FOR UPPER OR LOWER TAIL.

C234567
      SUBROUTINE MEANP(INO,IK,HYPD,ISUMTS,POBSH1,IUL,PEXIMP,PEX)
      PARAMETER(MAXT=270000)
      DOUBLE PRECISION HYPD(1000,0:2000),HYPD2(270000,20)
      DOUBLE PRECISION HYPD1(270000,1)
      DOUBLE PRECISION DENO,TS,POBSH1,PTOBS1,PTOBS2,PEXIMP,PEX
      DOUBLE PRECISION PEXIML,PEXIMU,PEXL,PEXU
      INTEGER INUM(270000,20),INUM1(270000,1)

      COMMON /DKIM/ DENO,ITOT,ISUML,INUM,HYPD2,INUM1,HYPD1


      OPEN(UNIT=30,FILE='pdis.out')


C      WRITE(30,100) INO
100   FORMAT('NO. =',I10)
C      WRITE(30,101) ISUMTS+ISUML,POBSH1/DENO
101   FORMAT('T =',I10,5x,'Pr(Table) =',F15.10)

C COMPUTATION OF LOWER TAIL P_VALUE
      IF (IUL .EQ. 1) GO TO 598
```

```
C-----------------------------------------------------------------
C  UPPER TAIL

C  TO MAKE INUM1 : ALL POSSIBLE TABLES SUCH THAT T>=t_obs
C  COMPUTATION OF PROB OF OBSERVING  RANDOM TABLES
C  TO MAKE HYPD1 : THE VALUE OF PROB FOR INUM1
C  PROB OF THE TABLE IS HYPD1(I,1)/DENO

      ICOUNT1=0
      DO 400 I=1,ITOT
      IF (INUM(I,IK+1) .GE. ISUMTS) THEN
       ICOUNT1=ICOUNT1+1
C       DO 410 J=1,IK+1
       INUM1(ICOUNT1,1)=INUM(I,IK+1)
       HYPD1(ICOUNT1,1)=HYPD2(I,IK+1)
C410    CONTINUE
      ENDIF
400   CONTINUE
C  INUM1 HAS T HYPD1 HAS Pr(Table).

C      WRITE(30,102) ICOUNT1
102   FORMAT('NO OF TABLES FOR T .GE. T_OBS =',I10)


      PTOBS1=0.D0
      PTOBS2=0.D0

      DO 560 I=1,ICOUNT1
      PTOBS2=PTOBS2+HYPD1(I,1)

      IF (INUM1(I,1) .GT. ISUMTS .OR. INUM1(I,1) .EQ. ISUMTS
     1    .AND. HYPD1(I,1) .LE. POBSH1) THEN
        PTOBS1=PTOBS1+HYPD1(I,1)
555   ENDIF
560   CONTINUE


      PEXIMU=PTOBS1/DENO
C  PEXIMPU IS IMPROVED UPPER TAIL EXACT PROB.

      PEXU=PTOBS2/DENO
C  PEXU  IS UPPER TAIL EXACT PROB.

      WRITE(30,903)INO,ISUMTS+ISUML,POBSH1/DENO,PEXIMU,PEXU
C      WRITE(30,901)PEXIMU
```

```
C      WRITE(30,902)PEXU
C      PRINT*
C     PRINT*,'PROB. OF OBSERVED TABLES =',POBSH1/DENO
C      PRINT*

      PEXIMP=PEXIMU
      PEXU=PEXU
      GO  TO 1000

888   FORMAT('       T      T_obs      Pr(Ta)       pr(Ta_obs)    P(T>T_obs+
     1order)')
889   FORMAT('       T      T_obs      Pr(Ta)       pr(Ta_obs)    P(T>=T_obs)
     2')

901   FORMAT('IMPROVED UPPER P_EXACT =',3X,F12.6)
902   FORMAT('   UPPER   P_EXACT     =',3X,F12.6,/)

c-------------------------------------------------------------------------

C----------------------------------------------------------------------
C  LOWER TAIL

C  TO MAKE INUM1 : ALL POSSIBLE TABLES SUCH THAT T<=t_obs
C  COMPUTATION OF PROB OF OBSERVING  RANDOM TABLES
C  TO MAKE HYPD1 : THE VALUE OF PROB FOR INUM1
C  PROB OF THE TABLE IS HYPD1(I,1)/DENO

598   ICOUNT1=0
      DO 600 I=1,ITOT
      IF (INUM(I,IK+1) .LE. ISUMTS) THEN
       ICOUNT1=ICOUNT1+1
C        DO 610 J=1,IK+1
       INUM1(ICOUNT1,1)=INUM(I,IK+1)
       HYPD1(ICOUNT1,1)=HYPD2(I,IK+1)
C610   CONTINUE
      ENDIF
600   CONTINUE

C      WRITE(30,103)ICOUNT1
103   FORMAT('NO OF TABLES FOR T .LE. T_OBS =',I10)


      PTOBS1=0.D0
      PTOBS2=0.D0
```

```
      DO 760 I=1,ICOUNT1
      PTOBS2=PTOBS2+HYPD1(I,1)

      IF (INUM1(I,1) .LT. ISUMTS .OR. INUM1(I,1) .EQ. ISUMTS
     1    .AND. HYPD1(I,1) .LE. POBSH1) THEN
        PTOBS1=PTOBS1+HYPD1(I,1)
755   ENDIF
760   CONTINUE

      PEXIML=PTOBS1/DENO
C  PEXIMPL IS IMPROVED LOWER TAIL EXACT PROB.

      PEXL=PTOBS2/DENO
C  PEXL IS LOWER TAIL EXACT PROB.

      WRITE(30,903)INO,ISUMTS+ISUML,POBSH1/DENO,PEXIML,PEXL

C      WRITE(30,903)PEXIML
C      WRITE(30,904)PEXL
C      PRINT*
C      PRINT*,'PROB. OF OBSERVED TABLES =',POBSH1/DENO
C      PRINT*

      PEXIMP=PEXIML
      PEX=PEXL

890   FORMAT('       T      T_obs      Pr(Ta)       pr(Ta_obs)   P(T<T_obs+
     1order)')
891   FORMAT('       T      T_obs      Pr(Ta)       pr(Ta_obs)   P(T<=T_obs)
     2')

903   FORMAT(I8,I7,1X,F15.10,1X,F12.6,F12.6)
904   FORMAT('   LOWER   P_EXACT     =',3X,F12.6,/)


1000  RETURN
      END

C**********************************************************************


C**********
C FROM KARIM MAY 90 SLR.FOR **********
      DOUBLE PRECISION FUNCTION SUMLG(DDD1,DDD2)
```

```
      DOUBLE PRECISION DDD,DD1,DD2,DDD1,DDD2
      DOUBLE PRECISION ZLOG,ZEXP,X
C
      ZLOG(X) = DLOG(X)
      ZEXP(X) = DEXP(X)
C
      DD1=DDD1
      DD2=DDD2
C     PRINT *,'HELLO FROM WITHIN SUMLG'
C
      DDD = DD1
      IF (DD2 .GT. DD1) DDD=DD2
      DD1 = ZEXP(DD1-DDD)
      DD2 = ZEXP(DD2-DDD)
      DDD = ZLOG(DD1+DD2) + DDD
      SUMLG = DDD
      RETURN
      END
C**********
C***********
C DIFFERENT SUMLG IN junE 91
c      DOUBLE PRECISION FUNCTION SUMLG(DD1,DD2)
c      DOUBLE PRECISION DDD,DD1,DD2
c      DOUBLE PRECISION ZLOG,ZEXP,X
cC
c      ZLOG(X) = DLOG(X)
c      ZEXP(X) = DEXP(X)
cC
c      DDD = DD1
c      IF (DD2 .GT. DD1) DDD=DD2
c      DD1 = ZEXP(DD1-DDD)
c      DD2 = ZEXP(DD2-DDD)
c      DDD = ZLOG(DD1+DD2) + DDD
c      SUMLG = DDD
c      RETURN
c      END
C*************************************************
      DOUBLE PRECISION FUNCTION FLOWER(BETA)
C
C  CALCULATES   P(T=T)*FF + P(T<T)
C
      DOUBLE PRECISION BETA,A(5500),SLA,SLB,SLU,UUU,SUMLG,C(5500),K,SLO
     +,FF
      INTEGER J,SCD
      COMMON/PARAM/C,J,SCD,K,FF
```

```
C
      DO 30 I=1,SCD
         A(I)=C(I)+(I-1)*BETA
   30 CONTINUE
C
      SLA=A(1)
      DO 40 I=2,J-1
        SLA=SUMLG(SLA,A(I))
   40 CONTINUE

      SLO=A(J)
      SLB=A(J+1)
      DO 50 I=J+2,SCD
         SLB=SUMLG(SLB,A(I))
   50 CONTINUE
      UUU=SUMLG(SLA,SLB)
      SLA=SUMLG(SLA,SLO+DLOG(FF))
      UUU=SUMLG(UUU,SLO)
C
      SLU = DEXP(SLA - UUU)
      SLU = SLU - K
      FLOWER = SLU
      RETURN
      END
C ***********************************
C**********************************************
      DOUBLE PRECISION FUNCTION FLOWBO(BETA)
C
C  CALCULATES   P(T=T)*FF WHEN T IS ON LOWER BOUNDARY
C
      DOUBLE PRECISION BETA,A(5500),SLA,SLB,SLU,UUU,SUMLG,C(5500),K,SLO
     +,FF
      INTEGER J,SCD
      COMMON/PARAM/C,J,SCD,K,FF
C
      DO 30 I=1,SCD
         A(I)=C(I)+(I-1)*BETA
   30 CONTINUE
C
      SLO=A(1)
      SLB=A(J+1)

      DO 50 I=J+2,SCD
         SLB=SUMLG(SLB,A(I))
   50 CONTINUE
```

```
      UUU=SUMLG(SLB,SLO)
      SLO=SLO+DLOG(FF)
C
      SLU = DEXP(SLO - UUU)
      SLU = SLU - K
      FLOWBO = SLU
      RETURN
      END
C ***********************************
C**********************************************
C**********************************************
      DOUBLE PRECISION FUNCTION FUPPER(BETA)
C
C  CALCULATES   P(T=T)*FF + P(T>T)
C
      DOUBLE PRECISION BETA,A(5500),SLA,SLB,SLU,UUU,SUMLG,C(5500),K,SLO
     +,FF
      INTEGER J,SCD
      COMMON/PARAM/C,J,SCD,K,FF

      DO 30 I=1,SCD
         A(I)=C(I)+(I-1)*BETA
   30 CONTINUE
C
      SLA=A(1)
      DO 40 I=2,J-1
        SLA=SUMLG(SLA,A(I))
   40 CONTINUE

      SLO=A(J)
      SLB=A(J+1)

      DO 50 I=J+2,SCD
         SLB=SUMLG(SLB,A(I))
   50 CONTINUE
      UUU=SUMLG(SLA,SLB)
      SLB=SUMLG(SLB,SLO+DLOG(FF))
      UUU=SUMLG(UUU,SLO)
      SLU = DEXP(SLB - UUU)
      SLU = SLU - K
      FUPPER = SLU
      RETURN
      END
C ***********************************
```

```
C***********************************************
      DOUBLE PRECISION FUNCTION FUPPBO(BETA)
C
C  CALCULATES   P(T=T)*FF  WHEN T IS ON UPPER BOUNDARY
C

      DOUBLE PRECISION BETA,A(5500),SLA,SLB,SLU,UUU,SUMLG,C(5500),K,SLO
     +,FF
      INTEGER J,SCD
      COMMON/PARAM/C,J,SCD,K,FF
C
      DO 30 I=1,SCD
         A(I)=C(I)+(I-1)*BETA
   30 CONTINUE
C
      SLA=A(1)
      DO 40 I=2,J-1
        SLA=SUMLG(SLA,A(I))
   40 CONTINUE

      SLO=A(J)

      UUU=SUMLG(SLA,SLO)
      SLO=SLO+DLOG(FF)
C
      SLU = DEXP(SLO - UUU)
      SLU = SLU - K
      FUPPBO = SLU
      RETURN
      END
C ************************************
C**********
      SUBROUTINE brent(X0,tol,IMAX,zbrent,Func,ITER)
c  FUNCTION ZBRENT(FUNC,X1,X2,TOL)
c  Van Wijngaarden-Dekker-Brent method
c  in Press WH, Flannery BP, Teukolsky SA, Vetterling WT:
c  Numerical Recipes - The Art of Scientific Computing
c  (Fortran version). Cambridge: Cambridge University Press, 1989
c  code on pages 253-254.
c
c  Using Brent's method, find the root of a function FUNC known to
c  lie between X1 and X2. The root returned as ZBRENT will be refined
c  until its accuracy is TOL.
c  (EPS is machine floating point precision, see p 16)
c
```

```
c  eps changed - declarations + delx0 introduced
c
      PARAMETER(ITMAX=100,EPS=1.d-14)
      double precision delx0,func,tol,zbrent,a,b,c,d,e,fa,fb,fc
      double precision p,q,r,s,xm,x0,tol1
      external func
      delx0=.2d+00
1     A=X0-delx0
      B=X0+delx0
      FA=FUNC(A)
      FB=FUNC(B)
      IF(FB*FA.GT.0)then
C           print *,'BRACKET ROOT||'
          delx0=delx0*2
          goto 1
      endif
c-------------------------no modifications below this line
      FC=FB
      DO 11 ITER=1,ITMAX
          IF(FB*FC.GT.0)THEN
            C=A
            FC=FA
            D=B-A
            E=D
          ENDIF
        IF(ABS(FC).LT.ABS(FB))THEN
            A=B
            B=C
            C=A
            FA=FB
            FB=FC
            FC=FA
        ENDIF
        TOL1=2.*EPS*ABS(B)+0.5*TOL
        XM=.5*(C-B)
          IF(ABS(XM).LE.TOL1 .OR. FB.EQ.0.)THEN
            ZBRENT=B
            RETURN
          ENDIF
          IF(ABS(E).GE.TOL1 .AND. ABS(FA).GT.ABS(FB))THEN
            S=FB/FA
            IF(A.EQ.C)THEN
              P=2.*XM*S
              Q=1.-S
            ELSE
```

```
              Q=FA/FC
              R=FB/FC
              P=S*(2.*XM*Q*(Q-R)-(B-A)*(R-1.))
              Q=(Q-1.)*(R-1.)*(S-1.)
            ENDIF
            IF(P.GT.0.) Q=-Q
              P=ABS(P)
              IF(2.*P .LT. MIN(3.*XM*Q-ABS(TOL1*Q),ABS(E*Q)))THEN
                E=D
                D=P/Q
              ELSE
                D=XM
                E=D
              ENDIF
          ELSE
            d=xm
            e=d
          endif
              A=B
              FA=FB
              IF(ABS(D) .GT. TOL1) THEN
                B=B+D
              ELSE
                B=B+SIGN(TOL1,XM)
              ENDIF
              FB=FUNC(B)
11        CONTINUE
          PAUSE 'MAX IT'
          ZBRENT=B
          RETURN
          end
C**********

C**********
      SUBROUTINE brent1(X0,tol,IMAX,zbrent,ITER,JCI,JCIO,PALPHA)

c  FUNCTION ZBRENT(FUNC,X1,X2,TOL)
c  Van Wijngaarden-Dekker-Brent method
c  in Press WH, Flannery BP, Teukolsky SA, Vetterling WT:
c  Numerical Recipes - The Art of Scientific Computing
c  (Fortran version). Cambridge: Cambridge University Press, 1989
c  code on pages 253-254.
c
c  Using Brent's method, find the root of a function FUNC known to
c  lie between X1 and X2. The root returned as ZBRENT will be refined
```

```
c  until its accuracy is TOL.
c  (EPS is machine floating point precision, see p 16)
c
c  eps changed - declarations + delx0 introduced
c
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER(ITMAX=100,EPS=1.d-14)
      double precision delx0,tol,zbrent,a,b,c,d,e,fa,fb,fc
C      double precision delx0,func,tol,zbrent,a,b,c,d,e,fa,fb,fc
      double precision p,q,r,s,xm,x0,tol1,OR1,OR2,FA1,FB1

      integer itab(1000,4),infhyl(1000),infhyu(1000)
      double precision hyp(0:2000),ds(0:1,0:5500),lge,POBSH
      DOUBLE PRECISION hypd(1000,0:2000),POBSH1,PEXIMP,PEX
      INTEGER J,SCD
      DOUBLE PRECISION CA(5500),K,FF

      COMMON/CI1/ik,mxs,mxz,mxd,lge,itab,hyp,ds,ipar,k1,k2,ierr,pobsh
      COMMON/CI2/hypd,infhyl,infhyu,POBSH1,PEXIMP,PEX
      COMMON/PARAM/CA,J,SCD,K,FF

C      external func
      delx0=1.2d0
      OR1=X0

1     IF (X0-delx0 .LE. 0.D0) THEN
          OR1=OR1/2.d0
       ELSE
          OR1=X0-delx0
       ENDIF

      OR2=X0+delx0
      A=OR1
      B=OR2
C      PRINT*
C      PRINT*,'ODDS RATIO OR1,OR2= ',OR1,OR2
      call cnv2x2(ik,mxs,mxz,mxd,lge,itab,hyp,ds,ipar,k1,k2,ierr,pobsh,
     1 jci,OR1)
      CALL IMPROV(ik,itab,hypd,infhyl,infhyu,POBSH1,PEXIMP,PEX,JCIO,OR1)
      FA=PEXIMP-K
      FA1=PEX-K

      call cnv2x2(ik,mxs,mxz,mxd,lge,itab,hyp,ds,ipar,k1,k2,ierr,pobsh,
     1 jci,OR2)
      CALL IMPROV(ik,itab,hypd,infhyl,infhyu,POBSH1,PEXIMP,PEX,JCIO,OR2)
```

```
      FB=PEXIMP-K
      FB1=PEX-K

      IF(FB*FA.GT.0)then
C           print *,'BRACKET ROOT||'
C           PRINT*,'K,X0,delx0,OR1,OR2,FA,FB',K,X0,delx0,OR1,OR2,FA,FB

           delx0=delx0*2
           goto 1
      endif
c------------------------no modifications below this line
      FC=FB
      DO 11 ITER=1,ITMAX
          IF(FB*FC.GT.0)THEN
            C=A
            FC=FA
            D=B-A
            E=D
          ENDIF
        IF(ABS(FC).LT.ABS(FB))THEN
            A=B
            B=C
            C=A
            FA=FB
            FB=FC
            FC=FA
        ENDIF
        TOL1=2.*EPS*ABS(B)+0.5*TOL
        XM=.5*(C-B)

          IF(ABS(XM).LE.TOL1 .OR. FB.EQ.0.)THEN
            ZBRENT=B
      OR2=B
C      PRINT*,'ODDS RATIO OR2 ',OR2

      call cnv2x2(ik,mxs,mxz,mxd,lge,itab,hyp,ds,ipar,k1,k2,ierr,pobsh,
     1 jci,OR2)
      CALL IMPROV(ik,itab,hypd,infhyl,infhyu,POBSH1,PEXIMP,PEX,JCIO,OR2)

            PALPHA=PEXIMP
C           PRINT*,'PEXIMP=',PEXIMP
C           PRINT*,'FIRST TETURN ZBRENT=B',B

            RETURN
          ENDIF
```

```
          IF(ABS(E).GE.TOL1 .AND. ABS(FA).GT.ABS(FB))THEN
            S=FB/FA
            IF(A.EQ.C)THEN
              P=2.*XM*S
              Q=1.-S
            ELSE
              Q=FA/FC
              R=FB/FC
              P=S*(2.*XM*Q*(Q-R)-(B-A)*(R-1.))
              Q=(Q-1.)*(R-1.)*(S-1.)
            ENDIF
            IF(P.GT.0.) Q=-Q
              P=ABS(P)
              IF(2.*P .LT. MIN(3.*XM*Q-ABS(TOL1*Q),ABS(E*Q)))THEN
                E=D
                D=P/Q
              ELSE
                D=XM
                E=D
              ENDIF
          ELSE
            d=xm
            e=d
          endif
              A=B
              FA=FB
              IF(ABS(D) .GT. TOL1) THEN
                B=B+D
              ELSE
                B=B+SIGN(TOL1,XM)
              ENDIF
C              FB=FUNC(B)
      OR2=B
C      PRINT*,'ODDS RATIO OR2 ',OR2

      call cnv2x2(ik,mxs,mxz,mxd,lge,itab,hyp,ds,ipar,k1,k2,ierr,pobsh,
     1 jci,OR2)
      CALL IMPROV(ik,itab,hypd,infhyl,infhyu,POBSH1,PEXIMP,PEX,JCIO,OR2)
      FB=PEXIMP-K
      FB1=PEX-K
11       CONTINUE
         PAUSE 'MAX IT'
         ZBRENT=B
C         PRINT*,'SECOND RETURN ZBRENT=B',B
```

```
        RETURN
        end
C**********

C**********
      SUBROUTINE SATO(ITAB,IK,LL,UL,MH,KA,IA,RLL,RUL,IPOS,VRBG)
C
C
C CALCULATES THE LIMITS OF EQUATION (2) IN
C SATO, T. (1990). CONFIDENCE LINITS FOR THE COMMON ODDS RATIO
C BASED ON THE ASYMPTOTIC DISTRIBUTION OF THE MANTEL-HAENSZEL
C HAENSZEL ESTIMATOR. BIOMETRICS, 46, 71-80.
C
C
      INTEGER ITAB(1000,4),IA,IPOS
      DOUBLE PRECISION IN,IM,INN,R,S,P,Q,W,RK,SK,SQ,LL,UL,CHI2,MH
      DOUBLE PRECISION KA(100,2),SVD1,SVD2,SVD3,VRBG,RUL,RLL
      DATA W/0.D00/,RK /0.D00/,SK /0.D00/,SVD1/0.D00/,SVD2/0.D00/
      DATA SVD3/0.D00/

      CHI2 = KA(IA,1)
C      PRINT *,CHI2

C  ADDED BY DONGUK KIM, OCT. 3, 1993
C  THIS IS REQUIRED FOR THE ITERATION OF RANDOM TABLES.
C  SET TO ZERO.

       W=0.D0
       RK=0.D0
       SK=0.D0
       SVD1=0.D0
       SVD2=0.D0
       SVD3=0.D0

      DO 100 I=1,IK
         IT = ITAB(I,1) + ITAB(I,2)
         IN = ITAB(I,1) + ITAB(I,3)
         IM = ITAB(I,2) + ITAB(I,4)
         INN = IN + IM
         R  = ITAB(I,1)*ITAB(I,4)/INN
         S  = ITAB(I,2)*ITAB(I,3)/INN
         P  = (ITAB(I,1) + ITAB(I,4))/INN
         Q  = (ITAB(I,2) + ITAB(I,3))/INN
         W  = W + (Q + 1/INN)*R + (P + 1/INN)*S
```

```
          RK = RK + R
          SK = SK + S
C--------------------------
C         ROBINS J, BRESLOW NE, GREENLAND S. ESTIMATORS OF
C         THE MANTEL-HAENSZEL VARIANCE CONSISTENT IN BOTH
C         SPARSE DATA AND LARGE-STRATA LIMITING MODELS
C         BIOMETRICS 1986;42:311-23.
C
C---------------------------RBG VARIANCE
          SVD1 = SVD1 + P*R
          SVD2 = SVD2 + (Q*R + P*S)
          SVD3 = SVD3 + Q*S
C-----------------------------------------
  100 CONTINUE
C--------------------------RBG LIMITS (CONT)
      IF(IPOS .GE. 1)THEN
            RUL=999
            RLL=999
            GOTO 109
      ENDIF
      VRBG=SVD1/2/RK/RK + SVD2/2/RK/SK + SVD3/2/SK/SK
      RLL = DEXP(DLOG(RK/SK)-SQRT(CHI2*VRBG))
      RUL = DEXP(DLOG(RK/SK)+SQRT(CHI2*VRBG))
C      PRINT *,RLL,RUL
C---------------------------------------------
 109  SQ = SQRT((4*RK*SK + CHI2*W)*CHI2*W)
      IF(SK .EQ. 0.0) GOTO 110
      LL = (2*RK*SK + CHI2*W - SQ)/2/SK/SK
      UL = (2*RK*SK + CHI2*W + SQ)/2/SK/SK
      MH = RK/SK
C      PRINT *,RK/SK
C      PRINT *,LL,UL
      GOTO 120
 110  LL = (2*RK*SK + CHI2*W - SQ)/2/RK/RK
      UL = (2*RK*SK + CHI2*W + SQ)/2/RK/RK
      LL = 1/UL
C      PRINT *,'INFINITE POINT ESTIMATE - LOWER LIMIT ONLY'
C      PRINT *,LL
 120  RETURN
      END
C**********


C234567
```

```
      SUBROUTINE IT2(ALPHA,INOUT,CA1,CA2,PT,ATO,SUM,IOOTO)
C
C  INOUT=1 IF t_obs IS IN THE GIVEN PROBABILITY DISTRIBUTION WITH
C  PROBABILITY 1-ALPHA, ELSE 0

      IMPLICIT REAL*8 (A-H,O-Z)
c      PARAMETER(EPS=1.d-6)
      PARAMETER(EPS=1.d-14)
      integer itab(1000,4),infhyl(1000),infhyu(1000)
      double precision hyp(0:2000),ds(0:1,0:5500),lge,POBSH
      DOUBLE PRECISION hypd(1000,0:2000),POBSH1,PEXIMP,PEX
      INTEGER J,SCD
      DOUBLE PRECISION CA(5500),K,FF,CA1(5500),CA2(5500),INC(5500)
      DOUBLE PRECISION CA3(5500)

      COMMON/CI1/ik,mxs,mxz,mxd,lge,itab,hyp,ds,ipar,k1,k2,ierr,pobsh
      COMMON/CI2/hypd,infhyl,infhyu,POBSH1,PEXIMP,PEX
      COMMON/PARAM/CA,J,SCD,K,FF

       DO 10 I=1,K2-K1+1
10     CA3(I)=CA1(I)

C  APPLYING MODIFIED P
       INOUT=1
       SUM=0.D0
       CALL SHELL(K2-K1+1,CA2)
       DO 100 I=1,K2-K1+1
       DO 110 I1=1,K2-K1+1
           IF (CA2(I) .EQ. CA3(I1)) THEN
            INC(I)=I1
C  FOR OTHER T THAT HAS THE SAME PROB.
            CA3(I1)=0.D0
            GO TO 100
           ENDIF
110   CONTINUE
c105        PRINT*,'CA2(I),INC(I)= ',CA2(I),INC(I)
100   CONTINUE

C
C  FOR TWO-SIDED LIMITS ADD TERMS FROM SMALLEST PROB
C  IN ASCENDING ORDER OF SIZE (NOT FROM EITHER TAIL).
C
      IP=1
150   SUM=SUM+CA2(IP)
      IIK=INC(IP)
```

```
      IF (IOOTO .EQ. 2) THEN
C         TWO SIDED-MODIFIED P (MODIFIED STERNE-TYPE P)
          IF (IIK .EQ. J) SUM=SUM-ATO
      ENDIF

      CA1(IIK)=0.D0
C      PRINT*,'SUM,IIK= ',SUM,IIK

      IF (SUM .GE. ALPHA) THEN
C        PRINT*,'SUM,IIK,INOUT= ',SUM,IIK,INOUT
        RETURN
      ENDIF

C      IF (INC(IP) .EQ. J) THEN
      IF (IP .EQ. K2-K1+1 .OR. CA2(IP+1) .GT. PT) THEN
          INOUT=0
C          PRINT*,'SUM,IIK,INOUT= ',SUM,IIK,INOUT
          RETURN
      ENDIF
      IP=IP+1
      GO TO 150
      END


C234567
      SUBROUTINE ITERA(ALPHA,START,RHO1,ist,PALPHA,IOOTO)
C  GIVEN ALPHA, STARTING VALUE, ITERA ITERATES AND RETURNS RHO A LOWER LIMIT.

      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER(EPS=1.d-14)
      PARAMETER(NNIT=1000)
c      PARAMETER(EPS=1.d-6)
      integer itab(1000,4),infhyl(1000),infhyu(1000)
      double precision hyp(0:2000),ds(0:1,0:5500),lge,POBSH,PSI
      DOUBLE PRECISION hypd(1000,0:2000),POBSH1,PEXIMP,PEX
      DOUBLE PRECISION SRT(NNIT,2),SRTS(NNIT),SRTSR(NNIT,2)
      DOUBLE PRECISION SRT1(NNIT,2),SRTS1(NNIT),SRTSR1(NNIT,2)
      INTEGER J,SCD
      DOUBLE PRECISION CA(5500),K,FF,CA1(5500),CA2(5500),CC(5500)

      COMMON/CI1/ik,mxs,mxz,mxd,lge,itab,hyp,ds,ipar,k1,k2,ierr,pobsh
      COMMON/CI2/hypd,infhyl,infhyu,POBSH1,PEXIMP,PEX
```

```
      COMMON/PARAM/CA,J,SCD,K,FF
      COMMON/ITN/SRT,SRTS,SRTSR,SRT1,SRTS1,SRTSR1

C      IF (IST .EQ. 1) THEN
C            OPEN(UNIT=35,FILE='st_lo_ci.out')
C            OPEN(UNIT=39,FILE='st_lo_all.out')
C      ELSE
C            OPEN(UNIT=36,FILE='st_up_ci.out')
C            OPEN(UNIT=40,FILE='st_up_all.out')
C      ENDIF


      DO 5 JJI=1,NNIT
      SRT(JJI,1)=0.D0
      SRT(JJI,2)=0.D0
      SRTS(JJI)=0.D0
      SRTSR(JJI,1)=0.D0
      SRTSR(JJI,2)=0.D0
      SRT1(JJI,1)=0.D0
      SRT1(JJI,2)=0.D0
      SRTS1(JJI)=0.D0
      SRTSR1(JJI,1)=0.D0
5     SRTSR1(JJI,2)=0.D0


      PSI=START
      RHO=0.D0
      KL=0
      ITE=0
      ITE1=0


C  FOR STERNE'S CI, JCI=1 SHOULD BE ASSIGNED FOR ODR COMPUTATION,
C  WHENEVER WE CALL CNV2X2.
      JCI=1
10    call cnv2x2(ik,mxs,mxz,mxd,lge,itab,hyp,ds,ipar,k1,k2,ierr,pobsh,
     1 jci,PSI)
      JCIO=3
      CALL IMPROV(ik,itab,hypd,infhyl,infhyu,POBSH1,PEXIMP,
     1 PEX,JCIO,PSI)

C  COMPUTE MODIFIED EXACT ALTERNATIVE PROB DISTN.
      CALL COMPT(CC,ATO,PT,IOOTO)
c      print*,'                          psi       =',psi
```

```
c  CA1(I) : PROBABILITY
      DO 90 I=1,K2-K1+1
      CA1(I)=CC(I)
90    CONTINUE

c  CA2(I) : DUPLICATE OF CA1 AND
C  THIS WILL BE SORTED PROBABILITY IN ASCENDING ORDER AFTER IT1.
      DO 95 I=1,K2-K1+1
      CA2(I)=CC(I)
95    CONTINUE


C        IN=INOUT(ALPHA)
C          CALL IT1(ALPHA,INOUT,CA1)
         CALL IT2(ALPHA,INOUT,CA1,CA2,PT,ATO,SUM,IOOTO)

         IN=INOUT
C
C    KL IS 0 UNTIL CORRECT VALUE IS SPANNED BY RHO AND OPSI,
C    THEN KL IS SET TO 1.
C
C    IN=1 IF PSI IS TOO LARGE, ELSE IN=0.
C
C  ATO IS INCLUDED IN ACCEPTANCE REGION.
       IF (CA1(J) .EQ. 0.D0) THEN
          CA1(J)=ATO
       ELSE
          CA1(J)=CA1(J)+ATO
       ENDIF


       PCHK=0.D0
       DO 100 I=1,K2-K1+1
       PCHK=PCHK+CA1(I)
C       IF (I .EQ. K2-K1+1) print*,i,CA1(I),pchk
100     continue
C       PRINT*,'PSI, TWO-SIDED P-VALUE =',PSI,SUM
C       PRINT*,'P_ACCEPT, TOTAL P =',PCHK,SUM+PCHK

          ITE1=ITE1+1
          SRT1(ITE1,1)=PSI
          SRT1(ITE1,2)=SUM

               IF (ITE1 .GT. 1000) THEN
                     PRINT*,'NOT CONVERGE IN TWO-SIDED P'
```

```
                        PALPHA=-99999.99999
                        RHO1=-99999.99999
                        GO TO 99
                 ENDIF


C  SUM IS THE TWO-SIDED P_VALUE.
      IF (IN .EQ. 0) THEN
           ITE=ITE+1
           SRT(ITE,1)=PSI
           SRT(ITE,2)=SUM
      ENDIF

      IF (KL .EQ. 1) GO TO 40
      IF (IN .EQ. 1) GO TO 20
      RHO=PSI
c      PSI=PSI*1.1D0
      if (ist .eq. 1) PSI=PSI*1.01D0
      if (ist .eq. 2) PSI=PSI*0.99D0


      GO TO 10
20    KL=1
      OPSI=PSI
30    PSI=(RHO+OPSI)*0.5D0
C
C   NEW ESTIMATE IS MIDPOINT OF SPANNING INTERVAL
      GO TO 10
40     IF (IN .EQ. 1) OPSI=PSI
       IF (IN .NE. 1) RHO=PSI
C       IF (DABS(RHO/OPSI -1.D0) .LT. EPS) RETURN

       IF (DABS(RHO/OPSI -1.D0) .LT. EPS) THEN

          IF (ITE1 .GT. NNIT) THEN
             PRINT*,'INCREASE NNIT FOR ARRAYS SRT,SRTS'
             GO TO 99
          ELSE
C               PRINT*,'NO OF ITERATION = ',ITE,ITE1
C               PRINT*
          ENDIF

      DO 102 JJI=1,ITE
102   SRTS(JJI)=SRT(JJI,2)
      CALL SHELL1(ITE,SRTS)
```

```
      DO 200 I=1,ITE
      DO 210 I1=1,ITE
            IF (SRTS(I) .EQ. SRT(I1,2)) THEN
                 SRTSR(I,1)=SRT(I1,1)
                 SRTSR(I,2)=SRT(I1,2)
                 GO TO 200
            ENDIF
210   CONTINUE
200   CONTINUE

C       IF (IST .EQ.1) THEN
C           DO 105 JJI=ITE,1,-1
C105     WRITE(35,107) JJI,SRTSR(JJI,1),SRTSR(JJI,2)
C       ELSE
C           DO 106 JJI=ITE,1,-1
C106     WRITE(36,107) JJI,SRTSR(JJI,1),SRTSR(JJI,2)
C       ENDIF
107   FORMAT(I10,2F20.15)

C         PALPHA=SRTSR(ITE,2)
C         RHO1=SRTSR(ITE,1)
C        PRINT*,'FINAL LIMIT(RHO) = ',RHO


C  SORTING BY THETA
      DO 300 JJI=1,ITE1
300   SRTS1(JJI)=SRT1(JJI,1)
      CALL SHELL1(ITE1,SRTS1)

C  FOR THE LOWER LIMIT P_VALUE IS SAVED IN ASCENDING ORDER.
      IF (IST .EQ. 1) THEN
      DO 310 I=1,ITE1
      DO 320 I1=1,ITE1
            IF (SRTS1(I) .EQ. SRT1(I1,1)) THEN
                 SRTSR1(I,1)=SRT1(I1,1)
                 SRTSR1(I,2)=SRT1(I1,2)
                 SRT1(I1,1)=0.D0
C               FOR THE SAKE OF THE SAME THETA.
                 GO TO 310
            ENDIF
320   CONTINUE
310   CONTINUE
C  FOR THE UPPER LIMIT P_VALUE IS SAVED IN ASCENDING ORDER.
C  THAT IS, THETA IS SAVED IN DESCENDING ORDER.
```

```
      ELSE
      DO 330 I=1,ITE1
      DO 340 I1=ITE1,1,-1
            IF (SRTS1(I) .EQ. SRT1(I1,1)) THEN
                 SRTSR1(ITE1-I+1,1)=SRT1(I1,1)
                 SRTSR1(ITE1-I+1,2)=SRT1(I1,2)
                 SRT1(I1,1)=0.D0
C               FOR THE SAKE OF THE SAME THETA.
                 GO TO 330
            ENDIF
340   CONTINUE
330   CONTINUE
      ENDIF

      DO 350 JJI=1,ITE1
      IF (SRTSR1(JJI,1) .EQ. RHO) THEN
            ITE2=JJI
            GO TO 360
      ENDIF
350   CONTINUE

360   CONTINUE
C360   PRINT*,'LIMIT(RHO) (ITE2) =',ITE2
      ITE3=ITE2
      DO 400 JJI=ITE2,1,-1
400   IF (SRTSR1(JJI,2) .GT. ALPHA) ITE3=JJI

      IF (ITE3 .NE. ITE2) THEN
             ITE4=ITE3-1
      ELSE
             ITE4=ITE3
      ENDIF
C      PRINT*,'ITE3,ITE4 =',ITE3,ITE4

C  FIND THE MAXIMUM P_VALUE WHICH CAN NOT EXCEED ALPHA/2 AND ITS THETA.
      TMAX=SRTSR1(ITE4,1)
      PRMAX=SRTSR1(ITE4,2)

      PALPHA=PRMAX
      RHO1=TMAX
C      PRINT*,'CLOSER ALPHA=',SRTSR(ITE,1),SRTSR(ITE,2)
C      PRINT*,'LIMIT(RHO) = ',TMAX,PRMAX

C       IF (IST .EQ.1) THEN
C           DO 420 JJI=ITE1,1,-1
```

```
C420     WRITE(39,107) JJI,SRTSR1(JJI,1),SRTSR1(JJI,2)
C       ELSE
C           DO 430 JJI=ITE1,1,-1
C430     WRITE(40,107) JJI,SRTSR1(JJI,1),SRTSR1(JJI,2)
C       ENDIF


99    RETURN
      ENDIF

       GO TO 30
       END


C234567
      SUBROUTINE COMPT(CC,ATO,PT,IOOTO)
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER(EPS=1.d-14)
      integer itab(1000,4),infhyl(1000),infhyu(1000)
      INTEGER INUM(270000,20),INUM1(270000,1)

      double precision hyp(0:2000),ds(0:1,0:5500),lge,POBSH
      DOUBLE PRECISION hypd(1000,0:2000),POBSH1,PEXIMP,PEX
      DOUBLE PRECISION HYPD1(270000,1),HYPD2(270000,20)
      DOUBLE PRECISION HYPD3(270000,1)
C  HYPD3(270000,1) IS PR(T) FOR EACH TABLE.

      INTEGER J,SCD
      DOUBLE PRECISION CA(5500),K,FF,CC(5500)
      DOUBLE PRECISION CHI(270000),CHI1(270000),CHIOBS


      COMMON/CI1/ik,mxs,mxz,mxd,lge,itab,hyp,ds,ipar,k1,k2,ierr,pobsh
      COMMON/CI2/hypd,infhyl,infhyu,POBSH1,PEXIMP,PEX
      COMMON/PARAM/CA,J,SCD,K,FF

      COMMON /DKIM/ DENO,ITOT,ISUML,INUM,HYPD2,INUM1,HYPD1
c      COMMON /DKIM1/ POBSH1,ISUMTS,IK
      COMMON /DKIM1/ ISUMTS
      COMMON / CH1/ CHI,CHIOBS


      DO 250 I=1,K2-K1+1
250   CC(I)=0.D0
```

```
      DO 300 I=1,ITOT
      IL=INUM(I,IK+1)+1
      CC(IL)=CC(IL)+HYPD2(I,IK+1)
300   CONTINUE

c      aa=0.d0
      DO 310 I=1,K2-K1+1
      CC(I)=CC(I)/DENO
c      aa=aa+cc(i)
c       print*,i,cc(i),aa
310    continue

       DO 320 I=1,ITOT
       IM=INUM(I,IK+1)+1
       HYPD3(I,1)=CC(IM)
320    CONTINUE

      ICOUNT1=0
      DO 400 I=1,ITOT
C      IF (INUM(I,IK+1) .EQ. ISUMTS) THEN
      IF (HYPD3(I,1) .EQ. CC(J)) THEN
       ICOUNT1=ICOUNT1+1
       INUM1(ICOUNT1,1)=INUM(I,IK+1)
       HYPD1(ICOUNT1,1)=HYPD2(I,IK+1)
       CHI1(ICOUNT1)=CHI(I)
      ENDIF
400   CONTINUE

      PTOBS3=0.D0
      PTOBS5=0.D0

      DO 560 I=1,ICOUNT1
C      IF (INUM1(I,1) .EQ. ISUMTS
C     1    .AND. HYPD1(I,1) .LE. POBSH1) THEN
C      IF (HYPD1(I,1) .LE. POBSH1) THEN

C  CHI-SQUARED STATISTIC IS USED FOR SECONDARY PARTITION.
      IF (CHI1(I) .GE. CHIOBS) THEN
        PTOBS3=PTOBS3+HYPD1(I,1)
      ELSE
        PTOBS5=PTOBS5+HYPD1(I,1)
      ENDIF
560   CONTINUE

      PTO=PTOBS3/DENO
```

```
      ATO=PTOBS5/DENO
      PT=CC(J)
C  ATO IS INCLUDED IN ACCEPTANCE REGION.

C       PRINT*,'IMPROVED P(T=T_O) = ',PTO
c       PRINT*,'ORD P - IMPROVED P(T=T_O) = ',ATO

C        CC(J)=PTO

       RETURN
       END



C234567
C  GIVEN ALPHA, STARTING VALUE, ITERA ITERATES AND RETURNS RHO A LIMIT.

      SUBROUTINE ITERA1(ALPHA,START,RHO1,ist,JCIO,PALPHA)

      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER(EPS=1.d-14)
      PARAMETER(NNIT=1000)
c      PARAMETER(EPS=1.d-6)
      integer itab(1000,4),infhyl(1000),infhyu(1000)
      double precision hyp(0:2000),ds(0:1,0:5500),lge,POBSH,PSI
      DOUBLE PRECISION hypd(1000,0:2000),POBSH1,PEXIMP,PEX
      DOUBLE PRECISION SRT(NNIT,2),SRTS(NNIT),SRTSR(NNIT,2)
      DOUBLE PRECISION SRT1(NNIT,2),SRTS1(NNIT),SRTSR1(NNIT,2)

      INTEGER J,SCD
      DOUBLE PRECISION CA(5500),K,FF

      COMMON/CI1/ik,mxs,mxz,mxd,lge,itab,hyp,ds,ipar,k1,k2,ierr,pobsh
      COMMON/CI2/hypd,infhyl,infhyu,POBSH1,PEXIMP,PEX
      COMMON/PARAM/CA,J,SCD,K,FF
      COMMON/ITN/SRT,SRTS,SRTSR,SRT1,SRTS1,SRTSR1

C       IF (IST .EQ. 1) THEN
C            OPEN(UNIT=33,FILE='mp_lo_ci.out')
C            OPEN(UNIT=37,FILE='mp_lo_all.out')
C       ELSE
C            OPEN(UNIT=34,FILE='mp_up_ci.out')
C            OPEN(UNIT=38,FILE='mp_up_all.out')
C       ENDIF
```

```
      DO 5 JJI=1,NNIT
      SRT(JJI,1)=0.D0
      SRT(JJI,2)=0.D0
      SRTS(JJI)=0.D0
      SRTSR(JJI,1)=0.D0
      SRTSR(JJI,2)=0.D0
      SRT1(JJI,1)=0.D0
      SRT1(JJI,2)=0.D0
      SRTS1(JJI)=0.D0
      SRTSR1(JJI,1)=0.D0
5     SRTSR1(JJI,2)=0.D0



      PSI=START
      RHO=0.D0
      KL=0
      ITE=0
      ITE1=0

      AALP=ALPHA/2.D0

C      IF (J .EQ. 1 .OR. J .EQ. SCD) AALP=ALPHA


C  FOR STERNE'S CI, JCI=1 SHOULD BE ASSIGNED FOR ODR COMPUTATION,
C  WHENEVER WE CALL CNV2X2.
      JCI=1
10    call cnv2x2(ik,mxs,mxz,mxd,lge,itab,hyp,ds,ipar,k1,k2,ierr,pobsh,
     1 jci,PSI)
C      JCIO=3
cc      JCIO=0
      CALL IMPROV(ik,itab,hypd,infhyl,infhyu,POBSH1,PEXIMP,
     1 PEX,JCIO,PSI)

          ITE1=ITE1+1
          SRT1(ITE1,1)=PSI
          SRT1(ITE1,2)=PEXIMP


                  IF (ITE1 .GT. 1000) THEN
                        PRINT*,'NOT CONVERGE IN ONE-SIDED MODIFIED P'
                        RHO1=-99999.99999
                        PALPHA=-99999.99999
```

```
                        GO TO 99
                   ENDIF


C      print*,'psi, PEXIMP =',psi,PEXIMP
C      IF (PEXIMP .GE. ALPHA/2.D0) THEN

      IF (PEXIMP .GE. AALP) THEN
           IN=1
      ELSE
           IN=0
           ITE=ITE+1
           SRT(ITE,1)=PSI
           SRT(ITE,2)=PEXIMP
      ENDIF

C
C    KL IS 0 UNTIL CORRECT VALUE IS SPANNED BY RHO AND OPSI,
C    THEN KL IS SET TO 1.
C
C    IN=1 IF PSI IS TOO LARGE, ELSE IN=0.
C

      IF (KL .EQ. 1) GO TO 40
      IF (IN .EQ. 1) GO TO 20
      RHO=PSI
      if (ist .eq. 1) PSI=PSI*1.01D0
      if (ist .eq. 2) PSI=PSI*0.99D0

      GO TO 10
20    KL=1
      OPSI=PSI
30    PSI=(RHO+OPSI)*0.5D0
C
C   NEW ESTIMATE IS MIDPOINT OF SPANNING INTERVAL
      GO TO 10
40     IF (IN .EQ. 1) OPSI=PSI
       IF (IN .NE. 1) RHO=PSI
c       IF (DABS(RHO/OPSI -1.D0) .LT. EPS) RETURN

       IF (DABS(RHO/OPSI -1.D0) .LT. EPS) THEN
      JCI=1
      PSI=RHO
      call cnv2x2(ik,mxs,mxz,mxd,lge,itab,hyp,ds,ipar,k1,k2,ierr,pobsh,
     1 jci,PSI)
```

```
      CALL IMPROV(ik,itab,hypd,infhyl,infhyu,POBSH1,PEXIMP,
     1 PEX,JCIO,PSI)
C       print*,'FINAL LIMIT : psi, PEXIMP =',psi,PEXIMP

      ITE=ITE+1
      SRT(ITE,1)=PSI
      SRT(ITE,2)=PEXIMP

          IF (ITE1 .GT. NNIT) THEN
             PRINT*,'INCREASE NNIT FOR ARRAYS SRT,SRTS'
             GO TO 99
          ELSE
C               PRINT*,'NO OF ITERATION ITE,ITE1= ',ITE,ITE1
C               PRINT*
          ENDIF

      DO 100 JJI=1,ITE
100   SRTS(JJI)=SRT(JJI,2)
      CALL SHELL1(ITE,SRTS)

      DO 200 I=1,ITE
      DO 210 I1=1,ITE
            IF (SRTS(I) .EQ. SRT(I1,2)) THEN
                 SRTSR(I,1)=SRT(I1,1)
                 SRTSR(I,2)=SRT(I1,2)
                 GO TO 200
            ENDIF
210   CONTINUE
200   CONTINUE

C       IF (IST .EQ.1) THEN
C           DO 105 JJI=ITE,1,-1
C105     WRITE(33,107) JJI,SRTSR(JJI,1),SRTSR(JJI,2)
C       ELSE
C           DO 106 JJI=ITE,1,-1
C106     WRITE(34,107) JJI,SRTSR(JJI,1),SRTSR(JJI,2)
C       ENDIF
107   FORMAT(I10,2F20.15)

C         PALPHA=SRTSR(ITE,2)
C         RHO1=SRTSR(ITE,1)


C  SORTING BY THETA
      DO 300 JJI=1,ITE1
```

```
 300   SRTS1(JJI)=SRT1(JJI,1)
       CALL SHELL1(ITE1,SRTS1)

 C  FOR THE LOWER LIMIT P_VALUE IS SAVED IN ASCENDING ORDER.
       IF (IST .EQ. 1) THEN
       DO 310 I=1,ITE1
       DO 320 I1=1,ITE1
             IF (SRTS1(I) .EQ. SRT1(I1,1)) THEN
                  SRTSR1(I,1)=SRT1(I1,1)
                  SRTSR1(I,2)=SRT1(I1,2)
                  SRT1(I1,1)=0.D0
 C               FOR THE SAKE OF THE SAME THETA.
                  GO TO 310
             ENDIF
 320   CONTINUE
 310   CONTINUE
 C  FOR THE UPPER LIMIT P_VALUE IS SAVED IN ASCENDING ORDER.
 C  THAT IS, THETA IS SAVED IN DESCENDING ORDER.
       ELSE
       DO 330 I=1,ITE1
       DO 340 I1=ITE1,1,-1
             IF (SRTS1(I) .EQ. SRT1(I1,1)) THEN
                  SRTSR1(ITE1-I+1,1)=SRT1(I1,1)
                  SRTSR1(ITE1-I+1,2)=SRT1(I1,2)
                  SRT1(I1,1)=0.D0
 C               FOR THE SAKE OF THE SAME THETA.
                  GO TO 330
             ENDIF
 340   CONTINUE
 330   CONTINUE
       ENDIF

       DO 350 JJI=1,ITE1
       IF (SRTSR1(JJI,1) .EQ. RHO) THEN
             ITE2=JJI
             GO TO 360
       ENDIF
 350   CONTINUE
 360   CONTINUE

 C      PRINT*,'LIMIT(RHO) (ITE2) =',ITE2
       ITE3=ITE2
       DO 400 JJI=ITE2,1,-1
 400   IF (SRTSR1(JJI,2) .GT. AALP) ITE3=JJI
 C400    IF (SRTSR1(JJI,2) .GT. ALPHA/2.D0) ITE3=JJI
```

```
      IF (ITE3 .NE. ITE2) THEN
             ITE4=ITE3-1
      ELSE
             ITE4=ITE3
      ENDIF
C      PRINT*,'ITE3,ITE4 =',ITE3,ITE4

C  FIND THE MAXIMUM P_VALUE WHICH CAN NOT EXCEED ALPHA/2 AND ITS THETA.
      TMAX=SRTSR1(ITE4,1)
      PRMAX=SRTSR1(ITE4,2)

      PALPHA=PRMAX
      RHO1=TMAX
C      PRINT*,'CLOSER ALPHA/2=',SRTSR(ITE,1),SRTSR(ITE,2)
C      PRINT*,'LIMIT(RHO) =',TMAX,PRMAX

C       IF (IST .EQ.1) THEN
C           DO 420 JJI=ITE1,1,-1
C420     WRITE(37,107) JJI,SRTSR1(JJI,1),SRTSR1(JJI,2)
C       ELSE
C           DO 430 JJI=ITE1,1,-1
C430     WRITE(38,107) JJI,SRTSR1(JJI,1),SRTSR1(JJI,2)
C       ENDIF


99    RETURN
      ENDIF

       GO TO 30
       END


*************************************************************************
*   SHELL SORT
*************************************************************************
C234567
      SUBROUTINE SHELL(N,ARR)
c       Sorts an array ARR of length N into ascending numerical order,
c       by the Shell-Mezgar algorithem (diminishing increment sort).
c       N is input; ARR is replaced on output by its sorted rearrangement.
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER (ALN2I=1.D0/0.69314718, TINY=1.E-5)
      REAL*8 ARR(5500)
      LOGNB2=INT(ALOG(FLOAT(N))*ALN2I+TINY)
```

```
      M=N
      DO 12 NN=1,LOGNB2
          M=M/2
          K=N-M
          DO 11 J=1,K
              I=J
3             CONTINUE
              L=I+M
              IF(ARR(L).LT.ARR(I)) THEN
                  T=ARR(I)
                  ARR(I)=ARR(L)
                  ARR(L)=T
                  I=I-M
                  IF(I.GE.1)GO TO 3
              ENDIF
11        CONTINUE
12    CONTINUE
      RETURN
      END

C234567
      SUBROUTINE SHELL1(N,ARR)
c       Sorts an array ARR of length N into ascending numerical order,
c       by the Shell-Mezgar algorithem (diminishing increment sort).
c       N is input; ARR is replaced on output by its sorted rearrangement.
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER (ALN2I=1.D0/0.69314718, TINY=1.E-5)
      PARAMETER (NNIT=1000)

      REAL*8 ARR(NNIT)
      LOGNB2=INT(ALOG(FLOAT(N))*ALN2I+TINY)
      M=N
      DO 12 NN=1,LOGNB2
          M=M/2
          K=N-M
          DO 11 J=1,K
              I=J
3             CONTINUE
              L=I+M
              IF(ARR(L).LT.ARR(I)) THEN
                  T=ARR(I)
                  ARR(I)=ARR(L)
                  ARR(L)=T
                  I=I-M
                  IF(I.GE.1)GO TO 3
```

```
              ENDIF
 11        CONTINUE
 12    CONTINUE
       RETURN
       END


       SUBROUTINE ITERA10(ALPHA,START,RHO1,ist,PALPHA,HYPD10)
C  GIVEN ALPHA, STARTING VALUE, ITERA ITERATES AND RETURNS RHO
C  A LOWER LIMIT.

       IMPLICIT REAL*8 (A-H,O-Z)
       PARAMETER(EPS=1.d-14)
       PARAMETER(NNIT=1000)
c       PARAMETER(EPS=1.d-6)
       integer itab(1000,4),infhyl(1000),infhyu(1000)
       double precision hyp(0:2000),ds(0:1,0:5500),lge,POBSH
       DOUBLE PRECISION hypd(1000,0:2000),POBSH1,PEXIMP,PEX
       DOUBLE PRECISION SRT(NNIT,2),SRTS(NNIT),SRTSR(NNIT,2)
       DOUBLE PRECISION SRT1(NNIT,2),SRTS1(NNIT),SRTSR1(NNIT,2)
       INTEGER J,SCD
       DOUBLE PRECISION CA(5500),K,FF,CA1(5500),CA2(5500),CC(5500)
       DOUBLE PRECISION HYPD2(270000,20),HYPD10(270000,1)


       COMMON/CI1/ik,mxs,mxz,mxd,lge,itab,hyp,ds,ipar,k1,k2,ierr,pobsh
       COMMON/CI2/hypd,infhyl,infhyu,POBSH1,PEXIMP,PEX
       COMMON/PARAM/CA,J,SCD,K,FF
       COMMON/ITN/SRT,SRTS,SRTSR,SRT1,SRTS1,SRTSR1


       PSI=START
       RHO=0.D0
       KL=0
       ITE=0
       ITE1=0


C  FOR STERNE'S CI, JCI=1 SHOULD BE ASSIGNED FOR ODR COMPUTATION,
C  WHENEVER WE CALL CNV2X2.
       JCI=1
 10    call cnv2x2(ik,mxs,mxz,mxd,lge,itab,hyp,ds,ipar,k1,k2,ierr,pobsh,
      1 jci,PSI)
       JCI0=3
       CALL IMPROV(ik,itab,hypd,infhyl,infhyu,POBSH1,PEXIMP,
```

```
     1 PEX,JCIO,PSI)

C  COMPUTE MODIFIED EXACT ALTERNATIVE PROB DISTN.
      CALL COMPT10(CC,ATO,PT,HYPD10)

      RETURN
      END

C234567
      SUBROUTINE COMPT10(CC,ATO,PT,HYPD10)
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER(EPS=1.d-14)
      integer itab(1000,4),infhyl(1000),infhyu(1000)
      INTEGER INUM(270000,20),INUM1(270000,1)

      double precision hyp(0:2000),ds(0:1,0:5500),lge,POBSH
      DOUBLE PRECISION hypd(1000,0:2000),POBSH1,PEXIMP,PEX
      DOUBLE PRECISION HYPD1(270000,1),HYPD2(270000,20)
      DOUBLE PRECISION HYPD3(270000,1),HYPD10(270000,1)

C  HYPD10(270000,1) IS PR(T) FOR EACH TABLE.

      INTEGER J,SCD
      DOUBLE PRECISION CA(5500),K,FF,CC(5500)

      COMMON/CI1/ik,mxs,mxz,mxd,lge,itab,hyp,ds,ipar,k1,k2,ierr,pobsh
      COMMON/CI2/hypd,infhyl,infhyu,POBSH1,PEXIMP,PEX
      COMMON/PARAM/CA,J,SCD,K,FF

      COMMON /DKIM/ DENO,ITOT,ISUML,INUM,HYPD2,INUM1,HYPD1
c      COMMON /DKIM1/ POBSH1,ISUMTS,IK
      COMMON /DKIM1/ ISUMTS


      DO 300 I=1,ITOT
       HYPD10(I,1)=HYPD2(I,IK+1)/DENO
300   CONTINUE


       RETURN
       END


C***********************************************************************
```

```
C       EFFICIENT SCORE TEST STATISTICS 4
C*********************************************************************
C234567
       SUBROUTINE CMHNN1(NROW,NCOL,NSTM,NIK,NJK,NTOT,MATRIX,CMH,G,
     1   JCIO,FIT)

C       TO COMPUTE THE EFFICIENT SCORE TEST STATISTIC.
C       MAX NO. OF STRATUM: 1000
C       NO. OF ROW AND COLUMN : 2, 2

C       COMMON IS USED FOR NIK,NJK,NTOT

C       IMPLICIT REAL*8 (A-H,O-Z)
      DOUBLE PRECISION X,EV,CMH,G
      DOUBLE PRECISION FIT(1000,2,2)
      INTEGER MATRIX(1000,2,2),NIK(2,1000),NJK(2,1000),NTOT(1000)

C       COMMON /A1/ NIK,NJK,NTOT

C       DO 90 K=1,NSTM
C       WRITE(*,1000)K,MATRIX(K,1,1),MATRIX(K,1,2),MATRIX(K,2,1),
C      1 MATRIX(K,2,2)
C       WRITE(*,1001)K,NIK(1,K),NIK(2,K),NJK(1,K),NJK(2,K),NTOT(K)
C90     CONTINUE
C1000  FORMAT('DATA',5I10)
C1001  FORMAT('TOTAL',6I10)

      X=0.D0
      G=0.D0

      DO 100 K=1,NSTM
      DO 110 I=1,NROW
      DO 110 J=1,NCOL

      IF (JCIO .EQ. 0) THEN
      EV=(NIK(I,K)*NJK(J,K))/DBLE(NTOT(K))
      ELSE
      EV=FIT(K,I,J)
      ENDIF

      X=X+((DBLE(MATRIX(K,I,J))-EV)**2)/EV

C       IF (MATRIX(K,I,J) .EQ. 0) GO TO 110
C       G=G+DBLE(MATRIX(K,I,J))*DLOG(DBLE(MATRIX(K,I,J))/EV)
110   CONTINUE
```

```
 100   CONTINUE

       CMH=X
 C      G=2.D0*G


 C      WRITE(*,1010)CMH,G
 C1010  FORMAT('CHI-SQUARED STATISTIC, G^2 =',2F12.7)

       RETURN
       END

C******************************************************************
C        ITERATIVE PROPORTIONAL FITTING ALGORITHM
C           (XZ,YZ) WITH N_{11K}=OR FOR K=1,IK, 1 OTHERWISE.
C******************************************************************
      SUBROUTINE IPF(PSI,IK,MATRIX,FIT)
C      IMPLICIT REAL*8(A-H,O-Z)
C  MAX NO. OF STRATA=100
C  MAX NO. OF ITERATION=2000

      PARAMETER(EPS=1.D-8)

      DOUBLE PRECISION X(2,2,100),E(2000,2,2,100),EE(3,2,100)
      DOUBLE PRECISION XX(3,2,100)
      DOUBLE PRECISION ETH(100),FIT(1000,2,2)
      DOUBLE PRECISION THETA,PSI,XA,EA,P1,P2,P3
      DOUBLE PRECISION FSIK(2,100),FSJK(2,100)


      INTEGER MATRIX(1000,2,2)


      THETA=PSI
      DO 5 K=1,IK
      DO 5 I=1,2
      DO 5 J=1,2
5     X(I,J,K)=DBLE(MATRIX(K,I,J))

      XA=0.D0
      DO 10 I=1,2
      DO 10 J=1,2
      DO 11 K=1,IK
11    XA=XA+X(I,J,K)
      XX(1,I,J)=XA
```

```
      XA=0.D0
 10   CONTINUE

      XA=0.D0
      DO 20 I=1,2
      DO 20 K=1,IK
      DO 21 J=1,2
 21   XA=XA+X(I,J,K)
      XX(2,I,K)=XA
      XA=0.D0
 20   CONTINUE

      XA=0.D0
      DO 30 J=1,2
      DO 30 K=1,IK
      DO 31 I=1,2
 31   XA=XA+X(I,J,K)
      XX(3,J,K)=XA
      XA=0.D0
 30   CONTINUE

C
C     SET TO 1 FOR INITIAL VALUE OF EXPECTED VALUE
C
      DO 40 I=1,2
      DO 40 J=1,2
      DO 50 K=1,IK
      IF (I .EQ. 1 .AND. J .EQ. 1) THEN
         E(1,I,J,K)=THETA
      ELSE
         E(1,I,J,K)=1.D0
      ENDIF
50    CONTINUE
C50    E(1,I,J,K)=1.D0
40    CONTINUE

C
C   COMPUTATION ROUTINE
C
      N=1
C      KK=1
      KK=2
2222  N=N+1
      IF (N .GT. 2000) THEN
         PRINT*,'INCREASE ARRAY E(2000,2,2,100)  IN IPF !'
```

```
          PRINT*,'IT DOES NOT CONVERGE WITHIN 2000 ITERATIONS.'
          GO TO 999
      ENDIF

      IF (KK .EQ. 1) GO TO 1000
      IF (KK .EQ. 2) GO TO 2000
      IF (KK .EQ. 3) GO TO 3000
C     STEP 1
1000  EA=0.D0
      DO 45 I=1,2
      DO 45 J=1,2
      DO 46 K=1,IK
46    EA=EA+E(N-1,I,J,K)
      EE(1,I,J)=EA
      EA=0.D0
45    CONTINUE
      DO 100 I=1,2
      DO 100 J=1,2
      DO 100 K=1,IK
      E(N,I,J,K)=XX(1,I,J)*E(N-1,I,J,K)/EE(1,I,J)
100   CONTINUE
      KK=KK+1
      GO TO 555
C
C     STEP 2
C
2000  EA=0.D0
      DO 57 I=1,2
      DO 57 K=1,IK
      DO 51 J=1,2
51    EA=EA+E(N-1,I,J,K)
      EE(2,I,K)=EA
      EA=0.D0
57    CONTINUE
      DO 101 I=1,2
      DO 101 J=1,2
      DO 101 K=1,IK
      E(N,I,J,K)=XX(2,I,K)*E(N-1,I,J,K)/EE(2,I,K)
101   CONTINUE
      KK=KK+1
      GO TO 555
C
C    STEP 3
C
3000  EA=0.D0
```

```
      DO 60 J=1,2
      DO 60 K=1,IK
      DO 61 I=1,2
61    EA=EA+E(N-1,I,J,K)
      EE(3,J,K)=EA
      EA=0.D0
60    CONTINUE
      DO 102 I=1,2
      DO 102 J=1,2
      DO 102 K=1,IK
      E(N,I,J,K)=XX(3,J,K)*E(N-1,I,J,K)/EE(3,J,K)
102   CONTINUE
C      KK=1
      KK=2
      GO TO 555
C
C      CHECK CONVERGENCE
C
555   DO 103 I=1,2
      DO 103 J=1,2
      DO 103 K=1,IK
      N1=N-1
      N2=N-2
      IF (N1 .LT. 0) N1=1
      IF (N2 .LT. 0) N2=1
      P1=DABS(E(N,I,J,K)-E(N1,I,J,K))
      P2=DABS(E(N,I,J,K)-E(N2,I,J,K))
      P3=DABS(E(N1,I,J,K)-E(N2,I,J,K))
      IF (P1 .GT. EPS .OR. P2 .GT. EPS .OR. P3 .GT. EPS)
     1  GO TO 2222
103   CONTINUE
      GO TO 1111
C
C     PRINT
C
1111   CONTINUE

      DO 666 K=1,IK
      DO 667 I=1,2
      DO 667 J=1,2
667   FIT(K,I,J)=E(N,I,J,K)
666   CONTINUE

C      WRITE(*,131)
C      WRITE(*,123) (((X(I,J,K),K=1,IK),J=1,2),I=1,2)
```

```
C      WRITE(*,132)
C      DO 157 I1=1,3
C      WRITE(*,124)((XX(I1,J,K),K=1,IK),J=1,2)
C157   CONTINUE
C      WRITE(*,133)
C      WRITE(*,134)
C      DO 77 JJ=1,N
C  ESTIMATED ORS FOR EACH STRATUM
C      DO 78 K=1,IK
C78    ETH(K)=E(JJ,1,1,K)*E(JJ,2,2,K)/(E(JJ,1,2,K)*E(JJ,2,1,K))

C      NN=JJ-1
C      WRITE(*,125)NN,(((E(JJ,I,J,K),K=1,IK),J=1,2),I=1,2),
C     1 (ETH(K),K=1,IK)
C77    CONTINUE
C      WRITE(*,135)N-1

C-----------------------------------------------------------------
C  CHECK IF OBSERVED AND FITTED FREQUENCIES  MATCH.
C  XX(2,I,K) : NIK(I,K)  <->FSIK(I,K)
C  XX(3,J,K) : NJK(J,K) <->FSJK(J,K)
C  FSIK(I,K),FSJK(J,K)

      DO 670 K=1,IK
      DO 680 I=1,2
680   FSIK(I,K)=0.D0
      DO 690 J=1,2
690   FSJK(J,K)=0.D0
670   CONTINUE

      DO 700 K=1,IK
      DO 710 I=1,2
      DO 710 J=1,2
710   FSIK(I,K)=FSIK(I,K)+FIT(K,I,J)

      DO 720 J=1,2
      DO 720 I=1,2
720   FSJK(J,K)=FSJK(J,K)+FIT(K,I,J)
700   CONTINUE

C      WRITE(*,140)
C      WRITE(*,124)((FSIK(I,K),K=1,IK),I=1,2)
C      WRITE(*,141)
C      WRITE(*,124)((FSJK(J,K),K=1,IK),J=1,2)
140   FORMAT(/,10X,'X-Z MARGINAL DATA FOR FITTED VALUES')
```

```
 141  FORMAT(/,10X,'Y-Z MARGINAL DATA FOR FITTED VALUES')

      DO 730 K=1,IK
      DO 730 I=1,2
      IF (DABS(XX(2,I,K)-FSIK(I,K)) .GT. EPS) THEN
      PRINT*,I,K,XX(2,I,K),FSIK(I,K),XX(2,I,K)-FSIK(I,K)
      PRINT*,I,K,' OBSERVED AND FITTED FREQUENCIES DOES NOT MATCH '
      PRINT*,'  IN X-Z MARGINAL TABLE.'
      ENDIF
 730  CONTINUE

      DO 740 K=1,IK
      DO 740 J=1,2
      IF (DABS(XX(3,J,K)-FSJK(J,K)) .GT. EPS) THEN
      PRINT*,J,K,XX(3,J,K),FSJK(J,K),XX(3,J,K)-FSJK(J,K)
      PRINT*,J,K,' OBSERVED AND FITTED FREQUENCIES DOES NOT MATCH '
      PRINT*,'  IN Y-Z MARGINAL TABLE.'
      ENDIF
 740  CONTINUE


 123  FORMAT(10(8F9.3,/))
 124  FORMAT(20(5F9.3,/))
 125  FORMAT(I3,1X,10(8F9.3,/))
 131  FORMAT(10X,'OUTPUT',/,10X,'DATA')
 132  FORMAT(/,10X,'MARGINAL DATA FOR EACH STEP')
 133  FORMAT(/,10X,'EXPECTED VALUE IN EACH ITERATION')
 134  FORMAT(6X,' M(111) M(112) M(121) M(122) M(211) M(212)',
     1   ' M(221) M(222)  OR1  OR2')
 135  FORMAT(10X,'CONVERGENCE IN ',I5,' ITERATIONS.')

 999  RETURN
      END
```

APPENDIX B
SOURCE CODE FOR SIMULATION

Following are program structure and part of FORTRAN source code for approximating exact inference about conditional association in $I \times J \times K$ contingency tables. It shows how the estimate of the ordinary or modified exact P-value for six tests can be constructed.

B.1 Program Structure

Important parameters are defined as follows.

NROW : Integer : input : number of rows in the observed matrix

NCOL : Integer : input : number of columns in the observed matrix

NSTM : Integer : input : number of strata in the observed matrix

NROWT1 : Integer array(50) : output : vector of row totals for the observed matrix at each stratum

NCOLT1 : Integer array(50) : output : vector of column totals for the observed matrix at each stratum

NROWT : Integer array(20,50) : output : NROWT1 is combined for all the strata

NCOLT : Integer array(20,50) : output : NCOLT1 is combined for all the strata

NTOT : Integer array(20) : output : vector of stratum totals for the observed table

JWORK : Integer array(50) : output : workspace

MATRIX1 : Integer array (50,50) : output : the randomly generated two-way table at each stratum

MX : Integer array(20,50,50) : input : the observed three-way table

MATRIX : Integer array(20,50,50) : output : the randomly generated three-way table

NCODE : Integer : input : select the type of tests of conditional independence

NRCM : Integer : input : (NROW-1)×(NCOL-1)

IDUM : Negative Integer : input : Seed

CMH : double precision : output : score statistic

Important subroutines are defined as follows.

Subroutine RCONT2
(NROW,NCOL,NSTM,NROWT1,NCOLT1,JWORK,MATRIX1,KEY,IFAULT,IDUM)
: Generate Two-Way random tables with given marginal totals

Subroutine COMPTOT(K,NROW,NCOL,MX,NROWT,NCOLT,NTOT)
: Compute row, column, and stratum totals

Double precision Function RAN1(IDUM)
: Uniform Random Number Generator, which is used in Subroutine RCONT2

Subroutine GETWTS(NROW,NCOL,WTR,WTC,NCODE)
: Get scores if ordinal variable is used

Subroutine CMHNN(NRCM,NROW,NCOL,NSTM,MATRIX,CMH)
: Compute score statistic assuming no three-factor interaction when both X and Y are nominal

Subroutine CMHNO(NROW,NCOL,NSTM,MATRIX,CMH)
: Compute score statistic assuming no three-factor interaction when X is nominal,

and Y is ordinal

Subroutine CMHOO(NROW,NCOL,NSTM,MATRIX,CMH)
: Compute score statistic assuming no three-factor interaction when both X and Y are ordinal

Subroutine CMHNN1(NRCM,NROW,NCOL,NSTM,MATRIX,CMH)
: Compute score statistic permitting three-factor interaction when both X and Y are nominal

Subroutine CMHNO1(NROW,NCOL,NSTM,MATRIX,CMH)
: Compute score statistic permitting three-factor interaction when X is nominal, and Y is ordinal

Subroutine CMHOO1(NROW,NCOL,NSTM,MATRIX,CMH)
: Compute score statistic permitting three-factor interaction when both X and Y are ordinal

Other subroutines are involved to compute inverse matrix, matrix multiplication, and Kronecker product multiplication.

B.2 Part of Source Code

```
PROGRAM THREEWAY
PARAMETER(lda=250)
PARAMETER(lda1=15)
PARAMETER(epsilon=1.0E-14)
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 D(50,50),D1(lda),VK(20,lda,lda),V(lda,lda)
REAL*8 DIV(lda),det(2)
```

```
      DIMENSION NROWT(20,50),NCOLT(20,50),MATRIX(20,50,50)
      DIMENSION JWORK(50),MX(20,50,50),NTOT(20),NNTOT(20)
      DIMENSION NROWT1(50),NCOLT1(50),MATRIX1(50,50)
      DIMENSION NROWT2(50),NCOLT2(50)
      DIMENSION NIK(50,20),NJK(50,20)
C      DIMENSION NIJ(50,50)

      REAL*8 FACT(25001),WTR(50),WTC(50)
      LOGICAL KEY
      LOGICAL LSP,LSM
      LOGICAL KIM
C      LOGICAL KEY1

      COMMON /B/ NROWM,NCOLM,FACT
C      COMMON /B1/ NIK,NJK

      COMMON /A1/ NIK,NJK,NTOT

      COMMON /A3/ D,D1,VK,V,DIV,det
      COMMON /A4/ WTR,WTC

C
      COMMON /TEMPRY/ HOP
C
      DATA MAXTOT /25000/
C
C
C************   Input Simulation Information *************
C
      WRITE(*,10000)
10000 FORMAT(3(/),T12,'***** LxL5 (version 8.0 -- 4/16/94) *****',/,
     1       /,T12,'SIX EFFICIENT SCORE STATISTICS',/,
     2          T12,'FOR TESTING CONDITIONAL INDEPENDENCE',/,
     3          T12,'OF THREE-WAY TABLES.',/)

      write(*,10001)
10001 FORMAT(T12,'   THIS PROGRAM CALCULATES',/,
     1       T12,'PRECISE ESTIMATES AND CONFIDENCE INTERVALS',/,
     2       T12,'FOR THE  MODIFIED EXACT P-VALUES.',/,
     3       T12,'THEY UTILIZE BOTH SCORE STATISTICS.',/)

      WRITE(*,45)
45    FORMAT(/,/,/,'ENTER NUMBER OF STRATUMS: ')
      READ(*,*) NSTM
      WRITE(*,50)
```

```
50    FORMAT(/,'ENTER NUMBER OF ROWS AND COLS: ')
      READ(*,*) NROW,NCOL

52    WRITE(*,55)
55    FORMAT(/,'ENTER CODE FOR TESTING: ',
     1 /,/,'  ASSUMING NO-THREE FACTOR INTERACTION :',
     1    /,/,'        1 : NOMINAL BY NOMINAL',
     2      /,'        2 : NOMINAL BY ORDINAL',
     3      /,'        3 : ORDINAL BY ORDINAL',/
     4 /,/,'  W/O ASSUMING NO-THREE FACTOR INTERACTION :',
     1    /,/,'        4 : NOMINAL BY NOMINAL',
     2      /,'        5 : NOMINAL BY ORDINAL',
     3      /,'        6 : ORDINAL BY ORDINAL',/)

      READ(*,*) NCODE

      IF (NCODE.EQ.1 .OR. NCODE.EQ.2 .OR. NCODE.EQ.3 .OR.
     1 NCODE.EQ.4 .OR. NCODE.EQ.5 .OR. NCODE.EQ.6) GO TO 57
      PRINT*,'PLEASE ENTER THE NUMBER (1 TO 6).'
      GO TO 52

57    IF (NCODE .EQ. 1 .OR. NCODE .EQ. 4) GO TO 60
      CALL GETWTS(NROW,NCOL,WTR,WTC,NCODE)

60    WRITE(*,75)
75    FORMAT(/,'ENTER OBSERVED TABLES FOR EACH STRATUM (ROW BY ROW):')
      DO 5 K=1,NSTM
      PRINT*,'STRATUM NUMBER =',K
      DO 10 I=1,NROW
        READ(*,*) (MX(K,I,J),J=1,NCOL)
10    CONTINUE
5     CONTINUE

      WRITE(*,80)
80    FORMAT(/,'ENTER NUMBER OF SIMULATION:')
      READ(*,*) NSIM

      WRITE(*,85)
85    FORMAT(/,'ENTER THE SEED (INTEGER) :')
      READ(*,*) ISEED
C
C*************end Input ********************************

      DO 20 K=1,NSTM
      CALL COMPTOT(K,NROW,NCOL,MX,NROWT,NCOLT,NTOT)
```

```
20    CONTINUE

      DO 23 K=1,NSTM
23    NNTOT(K)=NTOT(K)

      CALL SHELL(NSTM,NNTOT)
      NNTOTAL=NNTOT(NSTM)
      PRINT*
      PRINT*,'MAX OF NTOT =',NNTOTAL

      PRINT*
      PRINT*,'ROW TOTAL'
      DO 25 K=1,NSTM
      PRINT*,K,' :  ',(NROWT(K,I),I=1,NROW)
25    CONTINUE
      PRINT*

      PRINT*,'COLUMN TOTAL'
      DO 27 K=1,NSTM
      PRINT*,K,' :  ',(NCOLT(K,I),I=1,NCOL)
27    CONTINUE
      PRINT*

      PRINT*,'STRATUM TOTAL'
      PRINT*,(NTOT(K),K=1,NSTM)
      PRINT*
      PRINT*

      CALL MARTAB(NROW,NCOL,NSTM,NROWT,NCOLT,NIK,NJK)
      NRCM=(NROW-1)*(NCOL-1)

      CALL CPOBS(NROW,NCOL,NSTM,MX,NNTOTAL,P_OBS)
      POBS=P_OBS

      IF (NCODE.LE.3) THEN

      IF (NCODE.EQ.1) THEN
          CALL CMHNN(NRCM,NROW,NCOL,NSTM,MX,CMH)
          CMHOBS=CMH
C            PRINT*,'CMH,CMHOBS=',CMH,CMHOBS
          CALL CMHNN1(NROW,NCOL,NSTM,MX,CMH)
          CMHOBS1=CMH
C            PRINT*,'CMH,CMHOBS1=',CMH,CMHOBS1
      ELSE
          IF (NCODE.EQ.2) THEN
```

```
            CALL CMHNO(NROW,NCOL,NSTM,MX,CMH)
            CMHOBS=CMH
            CALL CMHNN(NRCM,NROW,NCOL,NSTM,MX,CMH)
            CMHOBS1=CMH
         ELSE
            CALL CMHOO(NROW,NCOL,NSTM,MX,CMH)
            CMHOBS=CMH
            CALL CMHNO(NROW,NCOL,NSTM,MX,CMH)
            CMHOBS1=CMH
         ENDIF
      ENDIF

      ELSE
       IF (NCODE.EQ.4) THEN
          CALL CMHNN1(NROW,NCOL,NSTM,MX,CMH)
          CMHOBS=CMH
C  NO MORE GENERAL STATISTIC FOR T'; WILL USE P({N})
       ELSE
           IF (NCODE.EQ.5) THEN
              CALL CMHNO1(NROW,NCOL,NSTM,MX,CMH)
              CMHOBS=CMH
              CALL CMHNN1(NROW,NCOL,NSTM,MX,CMH)
              CMHOBS1=CMH
           ELSE
              CALL CMHOO1(NROW,NCOL,NSTM,MX,CMH)
              CMHOBS=CMH
              CALL CMHNO1(NROW,NCOL,NSTM,MX,CMH)
              CMHOBS1=CMH
           ENDIF
       ENDIF
      ENDIF

      PRINT*
      PRINT*,'THE OBSERVED PRIMARY SCORE STATISTIC =',CMHOBS
      PRINT*,'THE OBSERVED SECONDRY SCORE STATISTIC =',CMHOBS1
      PRINT*,'PROB OF OBSERVED TABLE =',SNGL(POBS)
      PRINT*

      WRITE(*,28)
28    FORMAT(/,'PRINT EACH RANDOM TABLES ? (Y=1,N=0)')
      READ(*,*) NDATA

C********************** begin simulation ***********************

      S=0.D0
```

```
      ITCOUNT=0

 120  PRINT*
      PRINT*,'RANDOM TABLES WITH FIXED MARGINS'
      PRINT*

      ITCOUNT=ITCOUNT+1
      IDUM=(-1)*ISEED

      DO 100 IJ=1,NSIM

      SHOP=1.DO

      IF (NDATA .NE. 1) GO TO 103
      PRINT*,'SIMULATION NO = ',IJ

 103  DO 105 K=1,NSTM
      KK=K
      DO 90 L=1,NROW
      NROWT1(L)=NROWT(KK,L)
 90   CONTINUE
      DO 92 L=1,NCOL
      NCOLT1(L)=NCOLT(KK,L)
 92   CONTINUE

      CALL RCONT2(IJ,KK,NROW,NCOL,NSTM,NROWT1,NCOLT1,JWORK,
     1   MATRIX1,KEY,IFAULT,NNTOTAL,ISEED,IDUM)

C      PRINT*,'PROB OF RANDOM TABLE =',HOP
      SHOP=SHOP*HOP

      DO 95 L=1,NROW
      DO 96 M=1,NCOL
      MATRIX(KK,L,M)=MATRIX1(L,M)
 96   CONTINUE
 95   CONTINUE

      IF (NDATA .NE. 1) GO TO 97

      DO 98 I=1,NROW
      PRINT*,(MATRIX(KK,I,J),J=1,NCOL)
 98   CONTINUE
      PRINT*

 97   IF (IFAULT .NE. 0) THEN
```

```
          PRINT*,'IFAULT = ',IFAULT, 'KEY = ',KEY
          GO TO 1000
         ENDIF
 105     CONTINUE

C         CALL COMPMAR(NROW,NCOL,NSTM,MATRIX,NIJ,KEY1)
C         POBS : PROB OF OBSERVED TABLE, COMPUTED FROM CPOBS FOR TOTAL STRATUM
C         HOP  : PROB OF RANDOM TABLE,COMPUTED FROM RCONT2 FOR ONE STRATUM
C         SHOP : PROB OF RANDOM TABLE,COMPUTED FROM RCONT2 FOR TOTAL STRATUM
C

        IF (NCODE.LE.3) THEN

        IF (NCODE.EQ.1) THEN
            CALL CMHNN(NRCM,NROW,NCOL,NSTM,MATRIX,CMH)
            CMHRAN=CMH
            CALL CMHNN1(NROW,NCOL,NSTM,MATRIX,CMH)
            CMHRAN1=CMH
                IF (CMHRAN .GT. CMHOBS .OR.
     1             CMHRAN .EQ. CMHOBS .AND. CMHRAN1 .GE. CMHOBS1)
C     1             CMHRAN .EQ. CMHOBS .AND. SNGL(SHOP) .LE. SNGL(POBS))
     2          S=S+1

C            IF (CMHRAN .GE. CMHOBS) S=S+1
         ELSE
            IF (NCODE.EQ.2) THEN
                CALL CMHNO(NROW,NCOL,NSTM,MATRIX,CMH)
                CMHRAN=CMH
                CALL CMHNN(NRCM,NROW,NCOL,NSTM,MATRIX,CMH)
                CMHRAN1=CMH
                IF (CMHRAN .GT. CMHOBS .OR.
     1             CMHRAN .EQ. CMHOBS .AND. CMHRAN1 .GE. CMHOBS1)
C     1             CMHRAN .EQ. CMHOBS .AND. SNGL(SHOP) .LE. SNGL(POBS))
     2          S=S+1

C               IF (CMHRAN .GE. CMHOBS) S=S+1

           ELSE
                CALL CMHOO(NROW,NCOL,NSTM,MATRIX,CMH)
                CMHRAN=CMH
                CALL CMHNO(NROW,NCOL,NSTM,MATRIX,CMH)
                CMHRAN1=CMH
                IF (CMHRAN .GT. CMHOBS .OR.
     1             CMHRAN .EQ. CMHOBS .AND. CMHRAN1 .GE. CMHOBS1)
C     1             CMHRAN .EQ. CMHOBS .AND. SNGL(SHOP) .LE. SNGL(POBS))
```

```
     2        S=S+1

C              IF (CMHRAN .GE. CMHOBS) S=S+1
          ENDIF
      ENDIF

      ELSE

      IF (NCODE.EQ.4) THEN
          CALL CMHNN1(NROW,NCOL,NSTM,MATRIX,CMH)
          CMHRAN=CMH
C  NO MORE GENERAL STATISTIC FOR T'; USE P({Z})
              IF (CMHRAN .GT. CMHOBS .OR.
     1           CMHRAN .EQ. CMHOBS .AND. SNGL(SHOP) .LE. SNGL(POBS))
     2        S=S+1

C          IF (CMHRAN .GE. CMHOBS) S=S+1
       ELSE
          IF (NCODE.EQ.5) THEN
              CALL CMHNO1(NROW,NCOL,NSTM,MATRIX,CMH)
              CMHRAN=CMH
              CALL CMHNN1(NROW,NCOL,NSTM,MATRIX,CMH)
              CMHRAN1=CMH
              IF (CMHRAN .GT. CMHOBS .OR.
     1           CMHRAN .EQ. CMHOBS .AND. CMHRAN1 .GE. CMHOBS1)
C     1           CMHRAN .EQ. CMHOBS .AND. SNGL(SHOP) .LE. SNGL(POBS))
     2        S=S+1

C              IF (CMHRAN .GE. CMHOBS) S=S+1

          ELSE
              CALL CMHOO1(NROW,NCOL,NSTM,MATRIX,CMH)
              CMHRAN=CMH
              CALL CMHNO1(NROW,NCOL,NSTM,MATRIX,CMH)
              CMHRAN1=CMH
              IF (CMHRAN .GT. CMHOBS .OR.
     1           CMHRAN .EQ. CMHOBS .AND. CMHRAN1 .GE. CMHOBS1)
C     1           CMHRAN .EQ. CMHOBS .AND. SNGL(SHOP) .LE. SNGL(POBS))
     2        S=S+1

C              IF (CMHRAN .GE. CMHOBS) S=S+1
          ENDIF
      ENDIF

      ENDIF
```

```
      IF (NDATA .NE. 1) GO TO 100

      PRINT*,'THE PRIMARY SCORE STATISTIC FROM RANDOM TABLE =',CMHRAN
      PRINT*,'THE SECONDARY SCORE STATISTIC FROM RANDOM TABLE =',CMHRAN1
      PRINT*,'PROB OF RANDOM TABLE =',SNGL(SHOP)
      PRINT*,IJ,CMHRAN,CMHOBS,SNGL(S),SNGL(SHOP)
      PRINT*

 100  CONTINUE

      ITNSIM=NSIM*ITCOUNT

      P_EXACT=S/ITNSIM
      VAR_P=P_EXACT*(1.D0-P_EXACT)/ITNSIM
      CI1=P_EXACT-1.96D0*DSQRT(VAR_P)
      CI2=P_EXACT+1.96D0*DSQRT(VAR_P)
      STD_P=DSQRT(VAR_P)

      PRINT*
      PRINT*,'UPDATED ESTIMATE OF P_VALUE =',SNGL(P_EXACT)
      PRINT*,'UPDATED ESTIMATE OF VAR_P =',SNGL(VAR_P)
      PRINT*,'UPDATED ESTIMATE OF STD_P =',SNGL(STD_P)

      PRINT*
      PRINT*,'A 95% CONFIDENCE INTERVAL FOR  UPDATED ESTIMATE OF P :'
      PRINT*,'    ',SNGL(CI1),SNGL(CI2)

      WRITE(*,125) NSIM,ISEED
 125  FORMAT(/,I8,' TABLES SAMPLED WITH CURRENT STARTING  SEED',I8)
      WRITE(*,126) ITNSIM
 126  FORMAT(I8,' TABLES SAMPLED TOTALLY')

c*********************** end simulation *************************

      WRITE(*,110)
 110  FORMAT(/,'DO YOU WANT TO SAMPLE MORE TABLES ? (Y=1,N=0) :')
      READ(*,*) MORET

      IF (MORET .EQ. 1) THEN
         WRITE(*,115)
 115     FORMAT(/,'PLEASE REENTER THE SEED (INTEGER) :')
         READ(*,*) ISEED
         GO TO 120
      ENDIF
```

```
      PRINT*
      PRINT*,'END'

1000  STOP
      END


C**************** end of main program ************************

C234567
      SUBROUTINE RCONT2(IJ,KK,NROW,NCOL,NSTM,NROWT1,NCOLT1,JWORK,
     1   MATRIX1,KEY,IFAULT,NNTOTAL,ISEED,IDUM)
C
C        ALGORITHM AS 159 APPL. STATIST. (1981) VOL.30, NO.1
C        GENERATE RANDOM TWO-WAY TABLE WITH GIVEN MARGINAL TOTALS
C
C        CODES ARE MODIFIED BY DONGUK KIM TO BE USED FOR THE GENERATION
C        OF THREE WAY TABLES, AND DEXP,DBLE,AND DLOG ARE USED INSTEAD OF
C        EXP, FLOAT, AND LOG.
C        NNTOTAL IS THE MAXIMUM OF NTOTAL FOR THE STRATUM
C        AND USED IN COMPUTING LOG-FACTORALS.

      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION NROWT1(50),NCOLT1(50),MATRIX1(50,50)
      DIMENSION JWORK(50)
      DIMENSION NROWT2(50),NCOLT2(50)
      REAL*8 FACT(25001)
      LOGICAL KEY
      LOGICAL LSP,LSM
      COMMON /B/ NROWM,NCOLM,FACT
C
      COMMON /TEMPRY/ HOP
C
      DATA MAXTOT /25000/
C
C      IDUM=(KK+(IJ-1)*NSTM)*(-1)-ISEED
C      PRINT*,'IDUM=',IDUM

C
      IFAULT=0

      DO 100 I=1,NROW
      IF (NROWT1(I) .LE. 0) GOTO 214
100   CONTINUE
```

```
      NTOTAL=0
      DO 101 J=1,NCOL
      IF (NCOLT1(J) .LE. 0) GOTO 215
      NTOTAL=NTOTAL+NCOLT1(J)
 101  CONTINUE
      IF (NTOTAL .GT. MAXTOT) GOTO 216

      IF (KEY) GOTO 103
C
C        SET KEY FOR SUBSEQUENT CALLS
C
      KEY=.TRUE.
C
C        CHECK FOR FAULTS AND PREPARE FOR FUTURE CALLS
C
      IF (NROW .LE. 1) GOTO 212
      IF (NCOL .LE. 1) GOTO 213
      NROWM=NROW-1
      NCOLM=NCOL-1

C
C        CALCULATE LOG-FACTORIALS
C
      X=0.D0
      FACT(1)=0.D0
      DO 102 I=1,NNTOTAL
      X=X+DLOG(DBLE(I))
      FACT(I+1)=X
102   CONTINUE

c      print*,'I           factorial'
c      do 90 i=1,20
c      print*,i,dexp(fact(i))
c90    continue

C
C        -----------------------
C        CONSTRUCT RANDOM MATRIX
C        -----------------------
C
103   DO 105 J=1,NCOLM
105   JWORK(J)=NCOLT1(J)
      JC=NTOTAL
C
      HOP=1.D0
```

```
C
      DO 190 L=1,NROWM
      NROWTL=NROWT1(L)
      IA=NROWTL
      IC=JC
      JC=JC-NROWTL
      DO 180 M=1,NCOLM
      ID=JWORK(M)
      IE=IC
      IC=IC-ID
      IB=IE-IA
      II=IB-ID
C
C       TEST FOR ZERO ENTRIES IN MATRIX
C
      IF (IE .NE. 0) GOTO 130
      DO 121 J=M,NCOL
121   MATRIX1(L,J)=0
      GOTO 190
C
C         GENERATE PSEUDO-RANDUM NUMBER
C
130   RAND=RAN1(IDUM)
C
C        COMPUTE CONDITIONAL EXPECTED  VALUE OF MATRIX(L,M)
C
131   NLM=DBLE(IA*ID)/DBLE(IE)+0.5
      IAP=IA+1
      IDP=ID+1
      IGP=IDP-NLM
      IHP=IAP-NLM
      NLMP=NLM+1
      IIP=II+NLMP
      X=DEXP(FACT(IAP)+FACT(IB+1)+FACT(IC+1)+FACT(IDP)-
     1  FACT(IE+1)-FACT(NLMP)-FACT(IGP)-FACT(IHP)-FACT(IIP))
      IF (X .GE. RAND) GOTO 160
      SUMPRB=X
      Y=X
      NLL=NLM
      LSP=.FALSE.
      LSM=.FALSE.
C
C       INCREMENT ENTRY IN ROW L, COLUMN M
C
140   J=(ID-NLM)*(IA-NLM)
```

```
      IF (J .EQ. 0) GOTO 156
      NLM=NLM+1
      X=X*DBLE(J)/DBLE(NLM*(II+NLM))
      SUMPRB=SUMPRB+X
      IF (SUMPRB .GE. RAND) GOTO 160
150   IF (LSM) GOTO 155
C
C       DECREMENT ENTRY IN ROW L, COLUMN M
C
      J=NLL*(II+NLL)
      IF (J .EQ. 0) GOTO 154
      NLL=NLL-1
      Y=Y*DBLE(J)/DBLE((ID-NLL)*(IA-NLL))
      SUMPRB=SUMPRB+Y
      IF (SUMPRB .GE. RAND) GOTO 159
      IF (.NOT.LSP) GOTO 140
      GOTO 150
154   LSM=.TRUE.
155   IF (.NOT.LSP) GOTO 140
      RAND=SUMPRB*RAN1(IDUM)
      GOTO 131
156   LSP=.TRUE.
      GOTO 150
159   NLM=NLL
C
      HOP=HOP*Y
      GOTO 161
160   HOP=HOP*X
161   MATRIX1(L,M)=NLM
C
C160   MATRIX1(L,M)=NLM

      IA=IA-NLM
      JWORK(M)=JWORK(M)-NLM
180   CONTINUE
      MATRIX1(L,NCOL)=IA
190   CONTINUE
C
C       COMPUTE ENTRIES IN LAST ROW OF MATRIX
C
      DO 192 M=1,NCOLM
192   MATRIX1(NROW,M)=JWORK(M)
      MATRIX1(NROW,NCOL)=IB-MATRIX1(NROW,NCOLM)

C      PRINT*,'HOP = ',HOP
```

```
C
C       CHECK THE RANDOM TABLES SATISFY FIXED ROW TOTALS AND COLUMN TOTALS.
C
      CALL COMPTOT1(NROW,NCOL,MATRIX1,NROWT2,NCOLT2)
      DO 195 M=1,NROW
      IF (NROWT2(M) .NE. NROWT1(M)) GO TO 200
195   CONTINUE
      DO 197 M=1,NCOL
      IF (NCOLT2(M) .NE. NCOLT1(M)) GO TO 202
197   CONTINUE
      RETURN

C
C        SET FAULTS
C
212   IFAULT=1
      RETURN
213   IFAULT=2
      RETURN
214   IFAULT=3
      RETURN
215   IFAULT=4
      RETURN
216   IFAULT=5
      RETURN

200   PRINT*,M,'th ROW TOTAL IS WRONG.'
      RETURN
202   PRINT*,M,'th COLUMN TOTAL IS WRONG.'
      RETURN

      END


**********************************************************************
*    Uniform random generator
**********************************************************************

      DOUBLE PRECISION FUNCTION RAN1(IDUM)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 R(97)
      PARAMETER (M1=259200,IA1=7141,IC1=54773,RM1=3.8580247E-6)
      PARAMETER (M2=134456,IA2=8121,IC2=28411,RM2=7.4373773E-6)
```

```
      PARAMETER (M3=243000,IA3=4561,IC3=51349)
      DATA IFF /0/
      IF (IDUM.LT.0.OR.IFF.EQ.0) THEN
        IFF=1
        IX1=MOD(IC1-IDUM,M1)
        IX1=MOD(IA1*IX1+IC1,M1)
        IX2=MOD(IX1,M2)
        IX1=MOD(IA1*IX1+IC1,M1)
        IX3=MOD(IX1,M3)
        DO 11 J=1,97
          IX1=MOD(IA1*IX1+IC1,M1)
          IX2=MOD(IA2*IX2+IC2,M2)
          R(J)=(DBLE(IX1)+DBLE(IX2)*RM2)*RM1
11      CONTINUE
        IDUM=1
      ENDIF
      IX1=MOD(IA1*IX1+IC1,M1)
      IX2=MOD(IA2*IX2+IC2,M2)
      IX3=MOD(IA3*IX3+IC3,M3)
      J=1+(97*IX3)/M3
      IF(J.GT.97.OR.J.LT.1)PAUSE
      RAN1=R(J)
      R(J)=(DBLE(IX1)+DBLE(IX2)*RM2)*RM1
      RETURN
      END


C234567
      SUBROUTINE GETWTS(NROW,NCOL,WTR,WTC,NCODE)
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER NROW,NCOL
      REAL*8 WTR(50),WTC(50)
      IF(NCODE .EQ. 2 .OR. NCODE .EQ.5) GO TO 105
      WRITE(*,100)
100   FORMAT(/,'ENTER ROW SCORES: ')
      READ(*,*)(WTR(I),I=1,NROW)
105   WRITE(*,110)
110   FORMAT(/,'ENTER COLUMN SCORES: ')
      READ(*,*)(WTC(J),J=1,NCOL)

      RETURN
      END
```

```
C**********************************************************************
C       SCORE STATISTICS 1
C**********************************************************************

C234567
      SUBROUTINE CMHNN(NRCM,NROW,NCOL,NSTM,MATRIX,CMH)

C      TO COMPUTE SCORE STATISTIC
C      FOR THE TEST OF THE CONDITIONAL INDEPENDENCE OF THE I*J*K TABLES
C      WHEN X IS NOMINAL AND Y IS NOMINAL.
C      COMMON IS USED FOR NIK,NJK,NTOT

      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER(lda=250)
      DIMENSION MATRIX(20,50,50),NIK(50,20),NJK(50,20),NTOT(20)
      REAL*8 D(50,50),D1(lda),VK(20,lda,lda),V(lda,lda)
      REAL*8 DIV(lda),det(2)
c      REAL*8 D(50,50),D1(NRCM),VK(20,NRCM,NRCM),V(NRCM,NRCM)
c      REAL*8 DIV(NRCM),det(2)

c     real*8 VINV(lda,lda)
      integer lda,NRCM,info,job
      LOGICAL KIM
      COMMON /A1/ NIK,NJK,NTOT
      COMMON /A3/ D,D1,VK,V,DIV,det

      NROWM=NROW-1
      NCOLM=NCOL-1

C      COMPUTE D((NROWM*NCOLM),1) VECTOR
      DO 100 I=1,NROWM
      DO 105 J=1,NCOLM
      D(I,J)=0.D0
      DO 110 K=1,NSTM
      D(I,J)=D(I,J)+(MATRIX(K,I,J)-(NIK(I,K)*NJK(J,K))/DBLE(NTOT(K)))

110   continue
105   CONTINUE
100   CONTINUE

      DO 115 I=1,NROWM
      DO 120 J=1,NCOLM
      K=(I-1)*NCOLM+J
120   D1(K)=D(I,J)
115   CONTINUE
```

```
      IF (KIM) GO TO 15
C
C         SET KIM FOR SUBSEQUENT CALLS
C
      KIM=.TRUE.

C      COMPUTE V((NROWM*NCOLM),(NROWM*NCOLM)) MATRIX

122   DO 125 K=1,NSTM
      L=0
        DO 130 I=1,NROWM
        DO 140 J=1,NCOLM
        L=L+1
        M=0
           DO 150 IP=1,NROWM
           DO 160 JP=1,NCOLM
              IND1=0
              IND2=0
              IF (I.EQ.IP) IND1=1
              IF (J.EQ.JP) IND2=1
              M=M+1
              VK(K,L,M)=NIK(I,K)*(IND1*NTOT(K)-NIK(IP,K))
     1                 *NJK(J,K)*(IND2*NTOT(K)-NJK(JP,K))/
     2                 DBLE(NTOT(K)*NTOT(K)*(NTOT(K)-1.D0))
160           CONTINUE
150           CONTINUE
140     CONTINUE
130     CONTINUE
125   CONTINUE

      DO 170 I=1,NRCM
      DO 180 J=1,NRCM
      V(I,J)=0.D0
      DO 190 K=1,NSTM
190   V(I,J)=V(I,J)+VK(K,I,J)
180   CONTINUE
170   CONTINUE

      WRITE(*,195)
195   FORMAT(/,'PRINT NULL COVARIANCE MATRIX ? (Y=1,N=0)')
      READ(*,*) NCOV
      IF (NCOV .NE. 1) GO TO 196

      print*
```

```
      PRINT*,'NULL COVARIANCE MATRIX :'
      DO 191 I=1,NRCM
      PRINT*,(SNGL(V(I,J)),J=1,NRCM)
191   CONTINUE

196   JOB=01
      n=NRCM
c      lda=NRCM
      CALL dpofa(V,lda,n,info)

      IF (INFO .NE. 0) THEN
        WRITE(*,99) INFO
99      FORMAT(/,'THE FACTORIZATION IS NOT COMPLETE.',/,
     1 'THE LEADING MINOR OF ORDER',I5,'IS NOT POSITIVE DEFINITE.')
      PRINT*
      ENDIF

      CALL dpodi(V,lda,n,det,job)
C     COMPUTE DETERMINENT AND INVERSE MATRIX OF
c     A CERTAIN REAL SYMMETRIC POSITIVE DEFINITE MATRIX.
C     ONLY FOR SYMMETRIC MATRIX !!
c     DPODI PRODUCES THE UPPER HALF OF INVERSE OF V.
C     RETURNED V IS THE VAR-COV MATRIX OF V.

      DO 5 I=2,n
      DO 6 J=1,I-1
6     V(I,J)=V(J,I)
5     CONTINUE

7     WRITE(*,198)
198   FORMAT(/,'PRINT INVERSE MATRIX OF NULL COV. MATRIX ? (Y=1,N=0)')
      READ(*,*) NINVC
      IF (NINVC .NE. 1) GO TO 15

      PRINT*
      PRINT*,'INVERSE MATRIX : '
      do 10 i=1,n
10    print*,(SNGL(V(i,j)),j=1,n)


15    CALL MULTVA(D1,V,NRCM,NRCM,DIV)
      CALL INNER(DIV,D1,NRCM,CMHV)
      CMH=CMHV
C      PRINT*,'SCORE STATISTIC =',CMH
```

```
      RETURN
      END

C***********************************************************************
C        SCORE STATISTIC 2
C***********************************************************************

C234567
      SUBROUTINE CMHNO(NROW,NCOL,NSTM,MATRIX,CMH)

C       TO COMPUTE SCORE STATISTIC
C       FOR THE TEST OF THE CONDITIONAL INDEPENDENCE OF THE I*J*K TABLES
C       WHEN X IS NOMINAL AND Y IS ORDINAL.
C       COMMON IS USED FOR NIK,NJK,NTOT
C       COMMON IS USED FOR WTR,WTC

      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER(lda=250)
      PARAMETER(lda1=15)
      DIMENSION MATRIX(20,50,50),NIK(50,20),NJK(50,20),NTOT(20)
      REAL*8 WTR(50),WTC(50)
C       global arrays
      REAL*8 PIK(50,20),PJK(50,20)
      REAL*8 NK(20,lda,1),MK(20,lda,1),VK(20,lda,lda)
      REAL*8 GK(20,15,15),VGK(20,15,15),G(15,15),VG(15,15),GT(15,15)
      REAL*8 BK(lda,lda),BKT(lda,lda),CK(15,15)
C       local arrays
      REAL*8 A(15,15),A1(15,15),C1(15,15),C2(15,15),C3(15,15)
      REAL*8 D(15,15),B(15),Y(15)
      REAL*8 C(lda,lda),GNMK(lda,lda),YK(lda,lda),V(lda,lda)
      REAL*8 GTVG(15,15)

      integer lda,lda1,NROWM,NNN,NRNC,info,job
      LOGICAL KIM

      COMMON /A1/ NIK,NJK,NTOT
      COMMON /A4/ WTR,WTC

      NNN=1
      NROWM=NROW-1
      NRNC=NROW*NCOL

      IF (KIM) GO TO 1000
C
C          SET KIM FOR SUBSEQUENT CALLS
```

```
C
      KIM=.TRUE.

C***********************************************************
C     COMPUTE NULL VAR-COV MATRIX VG(NROWM,NROWM)
C***********************************************************

      DO 100 K=1,NSTM
      DO 110 I=1,NROW
110   PIK(I,K)=DBLE(NIK(I,K))/DBLE(NTOT(K))
      DO 120 J=1,NCOL
120   PJK(J,K)=DBLE(NJK(J,K))/DBLE(NTOT(K))
100   CONTINUE

C------------------------------------------------------
C     COMPUTE Mk=E(Nk|H0),WHICH IS SAVED IN MK(K,NRNC,1)
C
      DO 200 K=1,NSTM

      DO 230 I=1,NROW
230   A(I,1)=PIK(I,K)
      DO 240 J=1,NCOL
240   A1(J,1)=PJK(J,K)

C***************************************************
      CALL DIRECTMM(A,A1,C,NROW,NNN,NCOL,NNN)
C***************************************************

      DO 250 I=1,NRNC
250   MK(K,I,1)=NTOT(K)*C(I,1)

      DO 255 I=1,15
      DO 256 J=1,15
      A(I,J)=0.D0
      A1(I,J)=0.D0
256   CONTINUE
255   CONTINUE

      DO 257 I=1,lda
      DO 257 J=1,NNN
257   C(I,J)=0.D0

200   CONTINUE
C
C----------------------------------------------
```

```
C-----------------------------------------------------
C     COMPUTE Var(Nk|H0),WHICH IS SAVED IN VK(K,I,J)
C
      DO 350 K=1,NSTM

      DO 260 J=1,NCOL
      B(J)=PJK(J,K)
      A(J,1)=PJK(J,K)
      A1(1,J)=PJK(J,K)
260   CONTINUE

      CALL MMULTM(A,A1,NCOL,NNN,NCOL,C1)
      CALL DIAG(B,NCOL,D)
      CALL MSUBTM(D,C1,NCOL,NCOL,C2)

      DO 265 I=1,15
265   B(I)=0.D0
      DO 267 I=1,15
      DO 268 J=1,15
      A(I,J)=0.D0
      A1(I,J)=0.D0
      C1(I,J)=0.D0
      D(I,J)=0.D0
268   CONTINUE
267   CONTINUE

      DO 270 I=1,NROW
      B(I)=PIK(I,K)
      A(I,1)=PIK(I,K)
      A1(1,I)=PIK(I,K)
270   CONTINUE

      CALL MMULTM(A,A1,NROW,NNN,NROW,C1)
      CALL DIAG(B,NROW,D)
      CALL MSUBTM(D,C1,NROW,NROW,C3)

C**************************************************
      CALL DIRECTMM(C3,C2,C,NROW,NROW,NCOL,NCOL)
c**************************************************

      DO 280 I=1,NRNC
      DO 290 J=1,NRNC
290   VK(K,I,J)=DBLE(NTOT(K)*NTOT(K))/DBLE(NTOT(K)-1)*C(I,J)
280   CONTINUE
```

```
      DO 300 I=1,15
      DO 310 J=1,15
      A(I,J)=0.D0
      A1(I,J)=0.D0
      C1(I,J)=0.D0
      C2(I,J)=0.D0
      C3(I,J)=0.D0
      D(I,J)=0.D0
310   CONTINUE
300   CONTINUE

      DO 320 I=1,lda
      DO 330 J=1,lda
330   C(I,J)=0.D0
320   CONTINUE

      DO 340 I=1,15
340   B(I)=0.D0

350   CONTINUE
C
C-------------------------------------------------------
C-----------------------------------------------------------------
C     COMPUTE SCORE MATRIX BK(NROWM,NRNC)=CK(1,NCOL)@RK(NROWM,NROW)

      DO 400 I=1,NROWM
400   Y(I)=1.D0
      YY=-1.D0
      CALL AUGMD(Y,NROWM,YY,D)
C     D(NROWM,NROW)=RK
C     CK(1,NCOL) IS COLUMN SCORES.
      DO 410 J=1,NCOL
410   CK(1,J)=WTC(J)

C*************************************************
      CALL DIRECTMM(D,CK,BK,NROWM,NROW,NNN,NCOL)
C*************************************************

      CALL TRANS(BK,NROWM,NRNC,BKT)
C     BKT(NRNC,NROWM) IS TRANSPOSE OF BK(NROWM,NRNC).

      DO 446 I=1,15
446   Y(I)=0.D0
      DO 447 I=1,15
```

```
      DO 447 J=1,15
447   D(I,J)=0.D0

C---------------------------------------------------------------
C     COMPUTE VG(NROWM,NROWM). THIS IS SUMMING VGK(K,NROWM,NROWM)
C     ,WHICH IS Bk(VAR(Nk|H0)Bk', OVER K STRATUM.
C
      DO 450 K=1,NSTM

      DO 460 I=1,NRNC
      DO 470 J=1,NRNC
470   GNMK(I,J)=VK(K,I,J)
460   CONTINUE

      CALL MMULTM1(BK,GNMK,NROWM,NRNC,NRNC,YK)
      CALL MMULTM1(YK,BKT,NROWM,NRNC,NROWM,V)

      DO 475 I=1,NROWM
      DO 480 J=1,NROWM
480   VGK(K,I,J)=V(I,J)
475   CONTINUE

      DO 485 I=1,lda
      DO 490 J=1,lda
      GNMK(I,J)=0.D0
      YK(I,J)=0.D0
      V(I,J)=0.D0
490   CONTINUE
485   CONTINUE

450   CONTINUE

      DO 530 I=1,NROWM
      DO 540 J=1,NROWM
      VG(I,J)=0.D0
      DO 550 K=1,NSTM
550   VG(I,J)=VG(I,J)+VGK(K,I,J)
540   CONTINUE
530   CONTINUE

C     VG(NROWM,NROWM) IS VAR-COV MATRIX.
C**********************************************************

      WRITE(*,600)
600   FORMAT(/,'PRINT NULL COVARIANCE MATRIX ? (Y=1,N=0)')
```

```
      READ(*,*) NCOV
      IF (NCOV .NE. 1) GO TO 620

      print*
      PRINT*,'NULL COVARIANCE MATRIX :'
      DO 610 I=1,NROWM
      PRINT*,(SNGL(VG(I,J)),J=1,NROWM)
610   CONTINUE

620   JOB=01
      n=NROWM
c      lda=NRCM
      CALL dpofa(VG,lda1,n,info)

      IF (INFO .NE. 0) THEN
        WRITE(*,699) INFO
699     FORMAT(/,'THE FACTORIZATION IS NOT COMPLETE.',/,
     1 'THE LEADING MINOR OF ORDER',I5,' IS NOT POSITIVE DEFINITE.')
      PRINT*
      ENDIF

      CALL dpodi(VG,lda1,n,det,job)

      DO 605 I=2,n
      DO 606 J=1,I-1
606   VG(I,J)=VG(J,I)
605   CONTINUE

7     WRITE(*,698)
698   FORMAT(/,'PRINT INVERSE MATRIX OF NULL COV. MATRIX ? (Y=1,N=0)')
      READ(*,*) NINVC
      IF (NINVC .NE. 1) GO TO 1000

      PRINT*
      PRINT*,'INVERSE MATRIX : '
      do 690 i=1,n
690   print*,(SNGL(VG(i,j)),j=1,n)

C*************************************************************
C     COMPUTE G(NROWM,1). THIS IS SUMMING GK(K,NROWM,1)
C     , WHICH IS Bk(Nk-Mk), OVER K STRATUM.
C     G(NROWM,1) DEPENDS ON DATA Nk.
C*************************************************************

1000  DO 1005 K=1,NSTM
```

```
      DO 1010 I=1,NROW
      DO 1020 J=1,NCOL
      IJ=(I-1)*NCOL+J
      NK(K,IJ,1)=MATRIX(K,I,J)
1020  CONTINUE
1010  CONTINUE

      DO 1030 I=1,NRNC
      GNMK(I,1)=NK(K,I,1)-MK(K,I,1)
C     NK(K,I,1) IS DEFINED AS REAL*8
1030  CONTINUE

C     ARRAYS ARE (lda,lda). MMULTM1 IS CALLED INSTEAD OF MMULTM.
      CALL MMULTM1(BK,GNMK,NROWM,NRNC,NNN,YK)

      DO 1040 I=1,NROWM
      GK(K,I,1)=YK(I,1)
1040  CONTINUE

      DO 1050 I=1,lda
      GNMK(I,1)=0.D0
      YK(I,1)=0.D0
1050  CONTINUE

1005  CONTINUE

      DO 1060 I=1,NROWM
      DO 1070 J=1,NNN
      G(I,J)=0.D0
      DO 1080 K=1,NSTM
1080  G(I,J)=G(I,J)+GK(K,I,J)
1070  CONTINUE
1060  CONTINUE

C*****************************************************************

C*****************************************************************
C
C     COMPUTE SCORE STATISTIC
C     CMH=G'(VG^-1)G
C
C*****************************************************************

C     COMPUTE TRANSPOSE OF G(NROWM,1)
```

```
      DO 1100 I=1,NROWM
 1100 GT(1,I)=G(I,1)

      CALL MMULTM(GT,VG,NNN,NROWM,NROWM,GTVG)

      DO 1200 I=1,15
      B(I)=0.D0
      Y(I)=0.D0
 1200 CONTINUE
      DO 1210 I=1,NROWM
      B(I)=GTVG(1,I)
      Y(I)=G(I,1)
 1210 CONTINUE
      CALL INNER1(B,Y,NROWM,CMHV)
      CMH=CMHV
C      PRINT*,'C-M-H STATISTIC =',CMH

      DO 1212 I=1,15
      B(I)=0.D0
      Y(I)=0.D0
 1212 CONTINUE
      DO 1215 I=1,15
      DO 1215 J=1,15
      GTVG(I,J)=0.D0
 1215 D(I,J)=0.D0

      RETURN
      END

C***********************************************************************
C        SCORE STATISTIC 3
C***********************************************************************

C234567
      SUBROUTINE CMHOO(NROW,NCOL,NSTM,MATRIX,CMH)

C      TO COMPUTE SCORE STATISTIC
C      FOR THE TEST OF THE CONDITIONAL INDEPENDENCE OF THE I*J*K TABLES
C      WHEN X IS ORDINAL AND Y IS ORDINAL.
C      COMMON IS USED FOR NIK,NJK,NTOT
C      COMMON IS USED FOR WTR,WTC

      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION MATRIX(20,50,50),NIK(50,20),NJK(50,20),NTOT(20)
      REAL*8 WTR(50),WTC(50)
```

```
      COMMON /A1/ NIK,NJK,NTOT
      COMMON /A4/ WTR,WTC

      T=0.D0
      TT=0.D0
      DO 100 K=1,NSTM
      DO 110 I=1,NROW
      DO 120 J=1,NCOL
      T=T+WTR(I)*WTC(J)*(MATRIX(K,I,J)-(NIK(I,K)*NJK(J,K))
     1  /DBLE(NTOT(K)))
c      T1=WTR(I)*WTC(J)*(MATRIX(K,I,J)-(NIK(I,K)*NJK(J,K))/DBLE(NTOT(K)))

c     LINEAR RANK STATISTICS
      TT=TT+WTR(I)*WTC(J)*MATRIX(K,I,J)
c      TT1=WTR(I)*WTC(J)*MATRIX(K,I,J)
c      PRINT*,'T1=',T1,'TT1=',TT1

120   CONTINUE
110   CONTINUE
100   CONTINUE

C     CMH=T
      CMH=TT

c      PRINT*,'T=',T,'TT=',TT

      RETURN
      END

C*********************************************************************
C       SCORE STATISTIC 4
C*********************************************************************
C234567
      SUBROUTINE CMHNN1(NROW,NCOL,NSTM,MATRIX,CMH)

C      TO COMPUTE SCORE STATISTIC
C      FOR THE TEST OF THE CONDITIONAL INDEPENDENCE OF THE I*J*K TABLES
C      WHEN X IS NOMINAL AND Y IS NOMINAL.
C      W/O ASSUMING NO-THREE FACTOR INTERACTION MODEL.

C      MAX NO. OF STRATUM: 10
C      MAX NO. OF ROW*COL : 250
C      COMMON IS USED FOR NIK,NJK,NTOT
```

```
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION MATRIX(20,50,50),NIK(50,20),NJK(50,20),NTOT(20)

      COMMON /A1/ NIK,NJK,NTOT

      X=0.D0
      DO 100 K=1,NSTM
      DO 110 I=1,NROW
      DO 110 J=1,NCOL
      EV=(NIK(I,K)*NJK(J,K))/DBLE(NTOT(K))
      X=X+((DBLE(MATRIX(K,I,J))-EV)**2)/EV
c      print*,nik(i,k),njk(j,k),ntot(k)
c      print*,matrix(k,i,j),ev,x
110   CONTINUE
100   CONTINUE
      CMH=x
c      print*,'SCORE STATISTIC = ',CMH

      RETURN
      END


C***********************************************************************
C      SCORE STATISTIC 5
C***********************************************************************
C234567
      SUBROUTINE CMHNO1(NROW,NCOL,NSTM,MATRIX,CMH)

C      TO COMPUTE SCORE STATISTIC
C      FOR THE TEST OF THE CONDITIONAL INDEPENDENCE OF THE I*J*K TABLES
C      WHEN X IS NOMINAL AND Y IS ORDINAL.
C      W/O ASSUMING NO-THREE FACTOR INTERACTION MODEL.

C      MAX NO. OF STRATUM: 10
C      MAX NO. OF ROW*COL : 250

C      COMMON IS USED FOR NIK,NJK,NTOT
C      COMMON IS USED FOR WTR,WTC

      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION MATRIX(20,50,50),NIK(50,20),NJK(50,20),NTOT(20)
      REAL*8 WTR(50),WTC(50)
      REAL*8 UV(100),VK(10),GK(100,250),D(100,2500),DT(2500,100)
      REAL*8 P(2500),DP(2500,2500),PP(2500,2500),SIGMA(2500,2500)
      REAL*8 DSIGMA(100,2500),COVG(100,100),DIV(100)
```

```
      LOGICAL KIM

      COMMON /A1/ NIK,NJK,NTOT
      COMMON /A4/ WTR,WTC
      COMMON /A5/ P,DP,PP,SIGMA

      NNN=1
      NRNC=NROW*NCOL
      KNRNC=NSTM*NRNC
      NROWM=NROW-1
      NRNK=(NROW-1)*NSTM

      NTOTAL=0
      DO 100 K=1,NSTM
100   NTOTAL=NTOTAL+NTOT(K)
c       print*,'ntotal=',ntotal

      L=0
      DO 200 K=1,NSTM
      DO 210 I=1,NROWM
      L=L+1
      UV(L)=0.D0
      DO 220 J=1,NCOL
      UV(L)=UV(L)+WTC(J)*(MATRIX(K,I,J)-
     1   DBLE(NIK(I,K)*NJK(J,K))/DBLE(NTOT(K)))
220   CONTINUE
      UV(L)=UV(L)/DBLE(NTOTAL)
210   CONTINUE
200   CONTINUE

      IF (KIM) GO TO 900
C
C         SET KIM FOR SUBSEQUENT CALLS
C
      KIM=.TRUE.

C  NULL ASYMPTOTIC COVARIANCE OF SCORES.
C  COMPUTE GK(NRNK,NRNC)

      DO 250 K=1,NSTM
      VK(K)=0.D0
      DO 270 J=1,NCOL
270   VK(K)=VK(K)+WTC(J)*NJK(J,K)
250   CONTINUE
```

```
      L=0
      DO 280 K=1,NSTM
      DO 290 I=1,NROWM
      L=L+1
      M=0
      DO 300 IP=1,NROW
      IND1=0
      IF (I .EQ. IP) IND1=1
      DO 310 JP=1,NCOL
      M=M+1
      GK(L,M)=(WTC(JP)*NTOT(K)-VK(K))*
     1  (NTOT(K)*IND1-NIK(I,K))/DBLE(NTOT(K)*NTOT(K))
310   CONTINUE
300   CONTINUE
290   CONTINUE
280   CONTINUE

C  COMPUTE D(NRNK,KNRNC)
      DO 320 I=1,NRNK
      DO 330 J=1,KNRNC
330   D(I,J)=0.d0
320   CONTINUE

      L=0
      DO 350 K=1,NSTM
      DO 360 I=1,NROWM
      L=L+1
      DO 370 IJ=1,NRNC
      M=(K-1)*NRNC+IJ
      D(L,M)=GK(L,IJ)
370   CONTINUE
360   CONTINUE
350   CONTINUE

C  COMPUTE SIGMA(KNRNC,KNRNC)=DIAG(P)-PP'
      L=0
      DO 400 K=1,NSTM
      DO 410 I=1,NROW
      DO 420 J=1,NCOL
      L=L+1
420   P(L)=DBLE(NIK(I,K)*NJK(J,K))/DBLE((NTOT(K)*NTOTAL))
410   CONTINUE
400   CONTINUE
```

```
C  P(KNRNC), DP(KNRNC,KNRNC)
      CALL DIAG1(P,KNRNC,DP)
      CALL CMULR(P,KNRNC,NNN,KNRNC,PP)

      DO 500 I=1,KNRNC
      DO 510 J=1,KNRNC
510   SIGMA(I,J)=DP(I,J)-PP(I,J)
500   CONTINUE

C  COMPUTE COV(G(P))=D SIGMA D'/NTOTAL

C  TRANSPOSE OF D(NRNK,KNRNC) : DT(KNRNC,NRNK)
      DO 550 I=1,NRNK
      DO 560 J=1,KNRNC
560   DT(J,I)=D(I,J)
550   CONTINUE

C  COMPUTE D(NRNK,KNRNC)*SIGMA(KNRNC,KNRNC)=DSIGMA(NRNK,KNRNC)
C            print*,'dsigma(NRNK,knrnc)'
       DO 600 I=1,NRNK
       DO 610 J=1,KNRNC
       DSIGMA(I,J)=0.D0
       DO 620 K=1,KNRNC
       DSIGMA(I,J)=DSIGMA(I,J)+D(I,K)*SIGMA(K,J)
620     CONTINUE
           if (dabs(dsigma(i,j)) .lt. 1.0d-15) dsigma(i,j)=0.d0
610     CONTINUE
600     CONTINUE

c         do 622 i=1,NRNK
c622       print*,(sngl(dsigma(i,j)),j=1,knrnc)

C  COMPUTE DSIGMA(NRNK,KNRNC)*DT(KNRNC,NRNK)=COVG(NRNK,NRNK)
c          print*,'covg(NRNK,NRNK)'
      DO 650 I=1,NRNK
      DO 660 J=1,NRNK
      COVG(I,J)=0.D0
      DO 670 K=1,KNRNC
670   COVG(I,J)=COVG(I,J)+DSIGMA(I,K)*DT(K,J)
660   CONTINUE
c         print*,(sngl(covg(i,j)),j=1,NRNK)
650   CONTINUE

C  COMPUTE ESTIMATE COV G(P)
      DO 700 I=1,NRNK
```

```
      DO 710 J=1,NRNK
710   COVG(I,J)=COVG(I,J)/DBLE(NTOTAL)
700   CONTINUE

      WRITE(*,720)
720   FORMAT(/,'PRINT NULL COVARIANCE MATRIX ? (Y=1,N=0)')
      READ(*,*) NCOV
      IF (NCOV .NE. 1) GO TO 760

      print*
      PRINT*,'NULL COVARIANCE MATRIX :'
      DO 750 I=1,NRNK
      PRINT*,(SNGL(COVG(I,J)),J=1,NRNK)
750   CONTINUE

760   JOB=01
      n=NRNK
      lda=100
      CALL dpofa(COVG,lda,n,info)

      IF (INFO .NE. 0) THEN
        WRITE(*,699) INFO
699     FORMAT(/,'THE FACTORIZATION IS NOT COMPLETE.',/,
     1 'THE LEADING MINOR OF ORDER',I5,' IS NOT POSITIVE DEFINITE.')
      PRINT*
      ENDIF

      CALL dpodi(COVG,lda,n,det,job)

      DO 800 I=2,n
      DO 810 J=1,I-1
810   COVG(I,J)=COVG(J,I)
800   CONTINUE

      WRITE(*,850)
850   FORMAT(/,'PRINT INVERSE MATRIX OF NULL COV. MATRIX ? (Y=1,N=0)')
      READ(*,*) NINVC
      IF (NINVC .NE. 1) GO TO 900

      PRINT*
      PRINT*,'INVERSE MATRIX : '
      do 860 i=1,n
860   print*,(SNGL(COVG(i,j)),j=1,n)

C  COMPUTE SCORE STATISTIC : UV' COVG^-1 UV
```

```
900   CALL MULTVA2(UV,COVG,NRNK,NRNK,DIV)
      CALL INNER2(DIV,UV,NRNK,CMHV)
      CMH=CMHV
C      PRINT*,'SCORE STATISTIC FOR RANDOM TABLE =',CMH

      RETURN
      END



C***********************************************************************
C        SCORE STATISTIC 6
C***********************************************************************
C234567
      SUBROUTINE CMH001(NROW,NCOL,NSTM,MATRIX,CMH)

C      TO COMPUTE SCORE TEST STATISTI
C      FOR THE TEST OF THE CONDITIONAL INDEPENDENCE OF THE I*J*K TABLES
C      WHEN X IS ORDINAL AND Y IS ORDINAL.
C      W/O ASSUMING NO-THREE FACTOR INTERACTION MODEL.

C      MAX NO. OF STRATUM: 10
C      MAX NO. OF ROW*COL : 250
C      COMMON IS USED FOR NIK,NJK,NTOT
C      COMMON IS USED FOR WTR,WTC

      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION MATRIX(20,50,50),NIK(50,20),NJK(50,20),NTOT(20)
      REAL*8 WTR(50),WTC(50)
      REAL*8 UV(10),UK(10),VK(10),GK(10,250),D(10,2500),DT(2500,10)
      REAL*8 P(2500),DP(2500,2500),PP(2500,2500),SIGMA(2500,2500)
      REAL*8 DSIGMA(10,2500),COVG(10,10),DIV(10)

      LOGICAL KIM

      COMMON /A1/ NIK,NJK,NTOT
      COMMON /A4/ WTR,WTC
      COMMON /A5/ P,DP,PP,SIGMA

      NNN=1
      NRNC=NROW*NCOL
      KNRNC=NSTM*NRNC

      NTOTAL=0
```

```
      DO 100 K=1,NSTM
100   NTOTAL=NTOTAL+NTOT(K)
c       print*,'ntotal=',ntotal

      DO 200 K=1,NSTM
      UV(K)=0.D0
      DO 210 I=1,NROW
      DO 220 J=1,NCOL
      UV(K)=UV(K)+WTR(I)*WTC(J)*(MATRIX(K,I,J)-
     1   DBLE(NIK(I,K)*NJK(J,K))/DBLE(NTOT(K)))
220   CONTINUE
210   CONTINUE
      UV(K)=UV(K)/DBLE(NTOTAL)
200   CONTINUE

      IF (KIM) GO TO 900
C
C         SET KIM FOR SUBSEQUENT CALLS
C
      KIM=.TRUE.

C  NULL ASYMPTOTIC COVARIANCE OF SCORES.
C  COMPUTE GK(NSTM,NRNC)

      DO 250 K=1,NSTM
      UK(K)=0.D0
      VK(K)=0.D0
      DO 260 I=1,NROW
260   UK(K)=UK(K)+WTR(I)*NIK(I,K)
      DO 270 J=1,NCOL
270   VK(K)=VK(K)+WTC(J)*NJK(J,K)

      IJ=0
      DO 280 I=1,NROW
      DO 290 J=1,NCOL
      IJ=IJ+1
      GK(K,IJ)=WTR(I)*WTC(J)-(WTR(I)*VK(K)
     1  +WTC(J)*UK(K))/DBLE(NTOT(K))
     2  +UK(K)*VK(K)/DBLE(NTOT(K)*NTOT(K))
290   CONTINUE
280   CONTINUE
250   CONTINUE

C  COMPUTE D(NSTM,KNRNC)
      DO 300 K=1,NSTM
```

```
      DO 305 IJ=1,KNRNC
305   D(K,IJ)=0.d0
300   CONTINUE

c        print*,'d(k,l)'
      DO 310 K=1,NSTM
      DO 320 IJ=1,NRNC
      L=(K-1)*NRNC+IJ
      D(K,L)=GK(K,IJ)
320   CONTINUE
310   CONTINUE

C  COMPUTE SIGMA(KNRNC,KNRNC)=DIAG(P)-PP'
      L=0
      DO 400 K=1,NSTM
      DO 410 I=1,NROW
      DO 420 J=1,NCOL
      L=L+1
420   P(L)=DBLE(NIK(I,K)*NJK(J,K))/DBLE((NTOT(K)*NTOTAL))
410   CONTINUE
400   CONTINUE

C  P(KNRNC), DP(KNRNC,KNRNC)
      CALL DIAG1(P,KNRNC,DP)
      CALL CMULR(P,KNRNC,NNN,KNRNC,PP)

      DO 500 I=1,KNRNC
      DO 510 J=1,KNRNC
510   SIGMA(I,J)=DP(I,J)-PP(I,J)
500   CONTINUE

C  COMPUTE COV(G(P))=D SIGMA D'/NTOTAL

C  TRANSPOSE OF D(NSTM,KNRNC) : DT(KNRNC,NSTM)
      DO 550 I=1,NSTM
      DO 560 J=1,KNRNC
560   DT(J,I)=D(I,J)
550   CONTINUE

C  COMPUTE D(NSTM,KNRNC)*SIGMA(KNRNC,KNRNC)=DSIGMA(NSTM,KNRNC)
c          print*,'dsigma(nstm,knrnc)'
       DO 600 I=1,NSTM
       DO 610 J=1,KNRNC
       DSIGMA(I,J)=0.D0
       DO 620 K=1,KNRNC
```

```
      DSIGMA(I,J)=DSIGMA(I,J)+D(I,K)*SIGMA(K,J)
c        print*,i,j,D(I,K),SIGMA(K,J),D(I,K)*SIGMA(K,J),DSIGMA(I,J)

620      CONTINUE
            if (dabs(dsigma(i,j)) .lt. 1.0d-15) dsigma(i,j)=0.d0
610      CONTINUE
600      CONTINUE

C   COMPUTE DSIGMA(NSTM,KNRNC)*DT(KNRNC,NSTM)=COVG(NSTM,NSTM)
c        print*,'covg(nstm,nstm)'
      DO 650 I=1,NSTM
      DO 660 J=1,NSTM
      COVG(I,J)=0.D0
      DO 670 K=1,KNRNC
670   COVG(I,J)=COVG(I,J)+DSIGMA(I,K)*DT(K,J)
660   CONTINUE
650   CONTINUE

C   COMPUTE ESTIMATE COV G(P)
      DO 700 I=1,NSTM
      DO 710 J=1,NSTM
710   COVG(I,J)=COVG(I,J)/DBLE(NTOTAL)
700   CONTINUE

      WRITE(*,720)
720   FORMAT(/,'PRINT NULL COVARIANCE MATRIX ? (Y=1,N=0)')
      READ(*,*) NCOV
      IF (NCOV .NE. 1) GO TO 760

      print*
      PRINT*,'NULL COVARIANCE MATRIX :'
      DO 750 I=1,NSTM
      PRINT*,(SNGL(COVG(I,J)),J=1,NSTM)
750   CONTINUE


760   JOB=01
      n=NSTM
      lda=10
      CALL dpofa(COVG,lda,n,info)

      IF (INFO .NE. 0) THEN
        WRITE(*,699) INFO
699     FORMAT(/,'THE FACTORIZATION IS NOT COMPLETE.',/,
     1 'THE LEADING MINOR OF ORDER',I5,' IS NOT POSITIVE DEFINITE.')
```

```
      PRINT*
      ENDIF

      CALL dpodi(COVG,lda,n,det,job)

      DO 800 I=2,n
      DO 810 J=1,I-1
810   COVG(I,J)=COVG(J,I)
800   CONTINUE

      WRITE(*,850)
850   FORMAT(/,'PRINT INVERSE MATRIX OF NULL COV. MATRIX ? (Y=1,N=0)')
      READ(*,*) NINVC
      IF (NINVC .NE. 1) GO TO 900

      PRINT*
      PRINT*,'INVERSE MATRIX : '
      do 860 i=1,n
860   print*,(SNGL(COVG(i,j)),j=1,n)

C  COMPUTE SCORE TEST STATISTIC : UV' COVG^-1 UV

900   CALL MULTVA1(UV,COVG,NSTM,NSTM,DIV)
      CALL INNER1(DIV,UV,NSTM,CMHV)
      CMH=CMHV
C       PRINT*,'SCORE STATISTIC FOR RANDOM TABLE =',CMH

      RETURN
      END
```

REFERENCES

Agresti, A. (1990). *Categorical Data Analysis.* Wiley, New York.

Agresti, A. (1992). A survey of exact inference for contingency tables. *Statistical Science, 7*, 131–177.

Agresti, A., Mehta, C. R., & Patel, N. R. (1990). Exact inference for contingency tables with ordered categories. *Journal of the American Statistical Association, 85*, 453–458.

Agresti, A., Wackerly, D., & Boyett, J. (1979). Exact conditional tests for cross-classifications: Approximation of attained significance levels. *Psychometrika, 44*, 75–83.

Baptista, J., & Pike, M. C. (1977). Exact two-sided confidence limits for the odds ratio in a 2×2 table. *Journal of the Royal Statistical Society, Ser. C, 26*, 214–220.

Barnard, G. A. (1947). Significance tests for 2×2 tables. *Biometrika, 34*, 123–138.

Barnard, G. A. (1990). Must clinical trials be large? The interpretation of P-values and the combination of test results. *Statistics in Medicine, 9*, 601–614.

Birch, M. W. (1964). The detection of partial association I: The 2×2 case. *Journal of the Royal Statistical Society, Ser. B, 26*, 313–324.

Birch, M. W. (1965). The detection of partial association II: The general case. *Journal of the Royal Statistical Society, Ser. B, 27*, 111–124.

Boyett, J. (1979). Random $R \times C$ tables with given row and column totals. *Journal of the Royal Statistical Society, Ser. C, 28*, 329–332.

Clogg, C. C. (1982). Some models for the analysis of association in multiway cross-classifications having ordered categories. *Journal of the American Statistical Association, 77*, 803–815.

Cochran, W. G. (1954). Some methods of strengthening the common χ^2 tests. *Biometrics, 10*, 417–451.

Cohen, A., & Sackrowitz, H. B. (1991). Tests for independence in contingency tables with ordered categories. *Journal of Multivariate Analysis, 36*, 56–67.

Cohen, A., & Sackrowitz, H. B. (1992). An evaluation of some tests of trend in contingency tables. *Journal of the American Statistical Association, 87*, 470–475.

Cox, D. R. (1970). *Analysis of Binary Data.* Chapman and Hall, London.

Cox, D. R., & Hinkley, D. V. (1974). *Theoretical Statistics.* Chapman and Hall, London.

Cox, M. A. A., & Plackett, R. L. (1980). Small samples in contingency tables. *Biometrika, 67*, 1–13.

Davison, A. C. (1988). Approximate conditional inference in generalized linear models. *Journal of the Royal Statistical Society, Ser. B, 50*, 445–461.

Eaton, M. L. (1970). A complete class theorem for multidimensional one-sided alternatives. *Annals of Mathematical Statistics, 41*, 1884–1888.

Fortuin, C. M., Ginibre, J., & Kasteleyn, P. W. (1971). Correlation inequalities on some partially ordered sets. *Communications in Mathematical Physics, 22*, 89–103.

Gart, J. J. (1970). Point and interval estimation of the common odds ratio in the combination of 2×2 tables with fixed marginals. *Biometrika, 57*, 471–475.

Hirji, K. F., Mehta, C. R., & Patel, N. R. (1987). Computing distributions for exact logistic regression. *Journal of the American Statistical Association, 82*, 1110–1117.

Karlin, S., & Rinott, Y. (1980). Classes of orderings of measures and related correlation inequalities. I. Multivariate totally positive distributions. *Journal of Multivariate Analysis, 10*, 467–498.

Kuritz, S. J., Landis, J. R., & Koch, G. G. (1988). A general overview of Mantel-Haenszel methods: Applications and recent developments. *Annual Review of Public Health, 9*, 123–160.

Lamport, L. (1986). LaTeX*: A Document Preparation System.* Addison-Wesley Publishing Company, Reading, Massachusetts.

Lancaster, H. O. (1961). Significance tests in discrete distributions. *Journal of the American Statistical Association, 56*, 223–234.

Landis, J. R., Heyman, E. R., & Koch, G. G. (1978). Average partial association in three-way contingency tables: A review and discussion of alternative tests. *International Statistical Review, 46*, 237–254.

Ledwina, T. (1978a). Admissible tests for exponential families with finite support. *Mathematische Operationsforschung und Statistik, Ser. Statistics, 9*, 105–118.

Ledwina, T. (1978b). Admissible tests for contingency tables and Poisson's distributions, II. *Mathematische Operationsforschung und Statistik, Ser. Statistics, 9*, 119–125.

Ledwina, T. (1984). A note on admissibility of some tests of independence against 'positive dependence' in $R \times C$ contingence table. *Mathematische Operationsforschung und Statistik, Ser. Statistics, 15*, 565–570.

Lehmann, E. L. (1986). *Testing Statistical Hypotheses* (second edition). Wiley, New York.

Mantel, N. (1963). Chi-square tests with one degree of freedom: extensions of the Mantel-Haenszel procedure. *Journal of the American Statistical Association, 58*, 690–700.

Mantel, N., & Byar, D. P. (1978). Marginal homogeneity, symmetry and independence. *Communications in Statistics, Ser. A*, *7*, 953–976.

Mantel, N., & Haenszel, W. (1959). Statistical aspects of the analysis of data from retrospective studies of disease. *Journal of National Cancer Institute*, *22*, 719–748.

Marshall, A. W., & Olkin, I. (1970). *Inequalities: Theory of Majorization and Its Applications.* Academic Press, New York.

Matthes, T. K., & Truax, D. R. (1967). Tests of composite hypotheses for the multivariate exponential family. *Annals of Mathematical Statistics*, *38*, 681–697.

Mehta, C. R., Patel, N. R., & Gray, R. (1985). Computing an exact confidence interval for the common odds ratio in several 2 by 2 contingency tables. *Journal of the American Statistical Association*, *80*, 969–973.

Mehta, C. R., Patel, N. R., & Senchaudhuri, P. (1988). Importance sampling for estimating exact probabilities in permutational inference. *Journal of the American Statistical Association*, *83*, 999–1005.

Mehta, C. R., & Walsh, S. J. (1992). Comparison of exact, mid-p, and Mantel-Haenszel confidence intervals for the common odds ratio across several 2×2 contingency tables. *The American Statistician*, *46*, 146–150.

Mood, A. M., Graybill, F. A., & Boes, D. C. (1974). *Introduction to the Theory of Statistics* (third edition). McGraw-Hill, New York.

Neyman, J. (1935). On the problem of confidence limits. *Annals of Mathematical Statistics*, *6*, 111–116.

Pagano, M., & Halvorsen, K. T. (1981). An algorithm for finding the exact significance levels of $r \times c$ contingency tables. *Journal of the American Statistical Association*, *76*, 931–934.

Patefield, W. M. (1981). An efficient method of generating random $R \times C$ tables with given row and column totals. *Journal of the Royal Statistical Society, Ser. C*, *30*, 91–97.

Patefield, W. M. (1982). Exact tests for trends in ordered contingency tables. *Applied Statistics*, *31*, 32–43.

Pierce, D. A., & Peters, D. (1992). Practical use of higher order asymptotics for multiparameter exponential families. *Journal of the Royal Statistical Society, Ser. B*, *54*, 701–737.

Plackett, R. L. (1981). *The Analysis of Categorical Data* (second edition). Griffin, London.

Rao, C. R. (1973). *Linear Statistical Inference and Its Applications* (second edition). Wiley, New York.

Robins, J., Breslow, N., & Greenland, S. (1986). Estimators of the Mantel-Haenszel variance consistent in both sparse data and large-strata limiting models. *Biometrics*, *42*, 311–323.

SAS. (1988). *SAS/STAT User's Guide* (Release 6.03 edition). SAS Institute Inc., Cary, NC.

Skovgaard, I. (1987). Saddlepoint expansions for conditional distributions. *Journal of the Applied Probability*, *24*, 875–887.

StatXact. (1991). *StatXact: Statistical Software for Exact Nonparametric Inference* (Version 2 edition). Cytel Software, Cambridge, MA.

Sterne, T. E. (1954). Some remarks on confidence or fiducial limits. *Biometrika*, *41*, 275–278.

Streitberg, B., & Roehmel, J. (1990). On tests that are uniformly more powerful than the Wilcoxon-Mann-Whitney test. *Biometrics*, *46*, 481–484.

Suissa, S., & Shuster, J. J. (1985). Exact unconditional sample sizes for the 2 by 2 binomial trial. *Journal of the Royal Statistical Society, Ser. A*, *148*, 317–327.

Thomas, D. G. (1975). Exact and asymptotic methods for the combination of 2×2 tables. *Computers and Biomedical Research*, *8*, 423–446.

Tritchler, D. (1984). An algorithm for exact logistic regression. *Journal of the American Statistical Association*, *79*, 709–711.

Verbeek, A., & Kroonenberg, P. M. (1985). A survey of algorithms for exact distributions of test statistics in $r \times c$ contingency tables with fixed margins. *Computational Statistics and Data Analysis*, *3*, 159–185.

Vollset, S. E., & Hirji, K. F. (1991). A microcomputer program for exact and asymptotic analysis of several 2×2 tables. *Epidemiology*, *2*, 217–220.

Vollset, S. E., Hirji, K. F., & Elashoff, R. M. (1991). Fast computation of exact confidence limits for the common odds ratio in a series of 2×2 tables. *Journal of the American Statistical Association*, *86*, 404–409.

Yates, F. (1984). Tests of significance for 2×2 contingency tables. *Journal of the Royal Statistical Society, Ser. A*, *147*, 426–463.

Zelen, M. (1971). The analysis of several 2×2 contingency tables. *Biometrika*, *58*, 129–137.

BIOGRAPHICAL SKETCH

Donguk Kim was born on October 26, 1959 in Pusan, Korea. He was awarded a Bachelor of Economics degree in statistics in 1983, from Sung Kyun Kwan University, Korea. He also received a Master of Economics degree in statistics in 1985, from Sung Kyun Kwan University. He came to graduate school at the University of Florida in spring 1989.

While working toward his Ph.D in statistics from the University of Florida, he also worked as a teaching assistant and a statistical consultant for the Division of Biostatistics. He has been a member of the American Statistical Association since 1990.

He is married and has one child. After graduation he looks forward to doing teaching and research.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Alan Agresti, Chairman
Professor of Statistics

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Mark C.K. Yang
Professor of Statistics

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Myron N. Chang
Associate Professor of Statistics

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Brett D. Presnell
Assistant Professor of Statistics

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

David C. Wilson
Professor of Mathematics

This dissertation was submitted to the Graduate Faculty of the Department of Statistics in the College of Liberal Arts and Sciences and to the Graduate School and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

August 1994

____
Dean, Graduate School